HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Electronics, Communications and Automation

Yrjö Häme

Cortex Reconstruction from Magnetic Resonance Images

Thesis submitted for examination for the degree of Master of Science in Technology

Espoo, 24.10.2008

Supervisor: Professor Risto Ilmoniemi
Instructor: M.Sc. Mika Pollari

HELSINKI UNIVERSITY OF TECHNOLOGY ABSTRACT OF MASTER'S THESIS
FACULTY OF ELECTRONICS, COMMUNICATIONS AND AUTOMATION

Author:	Yrjö Häme
Faculty:	Faculty of Electronics, Communications and Automation
Major subject:	Computer and Information Science
Minor subject:	Imaging Technology
Title:	Cortex Reconstruction from Magnetic Resonance Images
Title in Finnish:	Korteksin rekonstruktio magneettikuvista
Chair:	Tfy-99 Biomedical Engineering
Supervisor:	Professor Risto Ilmoniemi
Instructor:	M.Sc. Mika Pollari

Abstract:

Cortex reconstruction is the process of generating surface representations of the cerebral cortex corresponding to its anatomy. An accurate reconstruction has several important applications in neuroscientific research, disease assessment and neurosurgical operations. Recent progress in segmentation methods has enabled performing the task automatically or with limited interaction. However, serious challenges caused by limitations of the magnetic resonance imaging technique and complexity of anatomical structures still remain.

Previously published cortex segmentation and reconstruction methods were reviewed and one of them was successfully implemented in this work, with some modifications to improve efficiency and accuracy. The chosen method generates topologically correct reconstructions of three cortical surfaces, including the central surface in addition to the usual inner and outer surfaces. Fuzzy clustering is used to find an initial segmentation and a geometric deformable model is then fitted to the result. The method also includes a special stage for tight cortical fold opening and computation of a vector field for locating the central surface.

The reconstruction results were evaluated with a landmark accuracy study using an unprecedentedly large data set of 30 image volumes and a total of 8640 landmark points. The results showed high average accuracy and robustness for the method, but also some bias and occasional large errors. Comparison was made with previous landmark study results for the same method, showing mostly consistent error values. In addition, a visual observation was performed on a total of 82 image volumes and the method was found to produce successful reconstructions every time, but locally some unwanted behavior was noted.

The implemented method generates high-quality reconstructions reliably, but areas of improvement were also found for achieving even better performance.

Number of pages: 69 **Keywords:** Cortex, reconstruction, segmentation, clustering, level sets

Department fills:

Approved: **Library code:**

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ
ELEKTRONIIKAN, TIETOLIIKENTEEN JA AUTOMAATION TIEDEKUNTA

Tekijä:	Yrjö Häme
Tiedekunta:	Elektroniikan, tietoliikenteen ja automaation tiedekunta
Pääaine:	Informaatiotekniikka
Sivuaine:	Kuvatekniikka
Työn nimi:	Korteksin rekonstruktio magneettikuvista
Title in English:	Cortex Reconstruction from Magnetic Resonance Images
Professuuri:	Tfy-99 Lääketieteellinen tekniikka
Työn valvoja:	Professori Risto Ilmoniemi
Työn ohjaaja:	DI Mika Pollari

Tiivistelmä:

Korteksin rekonstruktiossa luodaan aivokuorta kuvaava ja sen anatomiaa vastaava pintamalli. Tarkalla rekonstruktiolla on useita sovellusalueita aivotutkimuksessa, sairauksien tunnistamisessa ja arvioinnissa sekä kirurgisissa operaatioissa. Segmentointimenetelmien viimeaikainen kehitys on mahdollistanut tehtävän automaattisen suorittamisen. Silti kuvantamismenetelmän rajoitukset ja aivokuoren monimutkainen rakenne tuottavat rekonstruktiomenetelmille suuria haasteita.

Tässä työssä tehtiin katsaus korteksin rekonstruktio- ja segmentointimenetelmiin ja yksi menetelmistä toteutettiin tarkkuutta lisäävillä muutoksilla. Valittu menetelmä tuottaa topologisesti anatomiaa vastaavan rekonstruktion kolmelle pinnalle, aivokuoren ulko- ja sisäpintojen lisäksi näiden välissä sijaitsevalle keskipinnalle. Menetelmä käyttää rekonstruktiossa sumeaa klusterointia rakenteiden alustavaan segmentointiin, jonka lopputulokseen sovitetaan muovautuva pintamalli. Lisäksi menetelmä sisältää vaiheet aivokuoren uurteiden paikantamiseksi sekä keskipintaa ohjaavan vektorikentän laskemiseksi.

Rekonstruktiotuloksille tehtiin referenssipisteisiin pohjautuva arviointi huomattavasti suuremmalla aineistolla kuin samalle menetelmälle aiemmin, sisältäen 30 kuvaa ja yhteensä 8640 referenssipistettä. Tulokset osoittivat menetelmän toimivan toivotulla tavalla ja tuottavan keskimäärin hyvin tarkkoja tuloksia, mutta myös systemaattista virhettä ja yksittäisiä suuria virhearvoja havaittiin. Tuloksia vertailtiin aikaisempiin tutkimuksiin samalle menetelmälle ja virhearvot olivat pääosin vastaavia. Lisäksi visuaalinen arviointi suoritettiin yhteensä 82 kuvalle ja menetelmä tuotti onnistuneen rekonstruktion joka kerralla, joskin paikallisia epätarkkuuksia esiintyi myös.

Toteutettu menetelmä tuottaa korkeatasoisia rekonstruktioita luotettavasti, mutta mahdollisia kehittämiskohteita löydettiin yhä paremman suorituskyvyn saavuttamiseksi.

Sivumäärä: 69 **Avainsanat:** Korteksi, aivokuori, rekonstruktio, segmentointi, klusterointi, muovautuva malli

Täytetään osastolla:

Hyväksytty: **Kirjasto:**

Contents

Preface

Medical image processing has benefited greatly from the rapid increase in computing power, opening up a vast range of new possibilities for algorithm development. In segmentation, accuracy and robustness of results have taken great leaps forward, while the need for tedious manual interaction has reduced. The cortex reconstruction method implemented in this work reflects the recent advances, effectively combining several modern approaches.

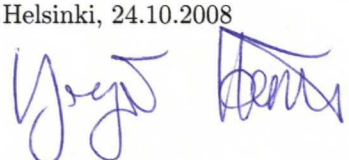
I could not have hoped for a better way to familiarize myself with the fascinating world of medical image processing than working on this thesis. Having no background in biomedical engineering, the work proved challenging at times, but also rewarding at the end. The outcome would naturally have been very different without all the help that I received in the process.

First and foremost, I thank my instructor and the head of our research group Mika Pollari for all the guidance and support that made my work possible. In addition, I am grateful that he got me in the group in the first place and inspired me to stay for future projects as well. I also thank my supervisor, professor Risto Ilmoniemi for his insightful comments that brought significant improvements to the thesis.

Special thanks to my colleague Mikko Lilja for all the invaluable instructions and practical help on anything I thought of asking. Thanks also to Jyrki Lötjönen at VTT for technical help and instructions, to Juhani Dabek for proofreading, and other colleagues at the Department of Biomedical Engineering and Computational Science.

Most importantly, I thank my parents and brothers for all the help and support they've given me through the years.

Helsinki, 24.10.2008



Yrjö Häme

Chapter 1

Introduction

Magnetic Resonance Imaging (MRI) provides high-quality information on soft tissues inside the human body (see Liew et al. [32]). Anatomically accurate segmentation and reconstruction of the cerebral cortex from magnetic resonance (MR) images is a difficult task but an important help in neuroscientific research and neurosurgical operations (see Tosun et al. [55]).

Neurodegenerative disorders, psychiatric disorders, a number of other diseases and healthy aging are often associated with structural changes in the brain. These changes can alter the imaging properties of brain tissues, as well as cause morphometric changes in brain structures. Morphometric alterations may include variations in the volume or shape of subcortical regions, or changes in the thickness, area and folding pattern of the cortex. Diseases typically diagnosed in this way include multiple sclerosis, schizophrenia, epilepsy, Alzheimer's disease, Parkinson's disease, band heterotopia and dysplasia. (See Fischl et al. [19], Suri et al. [51], Clarke et al. [11], Eskildsen et al. [18], Liew et al. [32]).

An accurate reconstruction of the cortex enables acquiring important information of the condition and location of brain structures, evaluating clinically relevant parameters such as surface area, structure, thickness and convolutedness of the cortex, as well as determining relationships between specific neuroanatomical structures. This information can be used for many purposes, such as quantitative assessment of brain diseases and treatment procedures, research in pathology prediction by determining morphological and structural changes or deformations, multi-modality fusion and registration, surgical planning, navigation and image-guided surgery, visualization, interactive segmentation, functional brain mapping and sub-cortical segmentation (see Suri et al. [51], Clarke et al. [11]).

The reconstructed cortex can be used in solving the inverse problem of e.g. magnetoencephalography (MEG) or electroencephalography (EEG) by combining the reconstruction with functional brain measurements to achieve an accurate localization of activity on the cortex. Electromagnetic imaging of the brain is the only functional imaging modality that can offer excellent time resolution. Unfortunately, the spatial resolution of these imaging methods is limited and different source arrangements can lead to the same measurements. Because of the non-uniqueness and numerical instability, the inverse problem of finding a

source map from data is very difficult. An exact reconstruction of the cortex can be used to impose anatomical constraints on the inverse problem. These constraints reduce the dimension of the search space, preventing solutions from being found in unrealistic locations. (See Baillet et al. [4]).

Manual segmentation of the cortex from MR images is laborious and inaccurate and the result varies strongly dependent on the observer. In addition, the result is not reproducible. These shortcomings have created a demand for automatic and semi-automatic segmentation methods, which should be fast, accurate and robust. (See Liew et al. [32], Suri et al. [51], Goldenberg et al. [20]).

Generally speaking, automatic segmentation of images is not a new area of research, and many algorithms have been developed and used in various applications. Segmentation of brain MR images has its own challenges, however. Intensity inhomogeneity, limited resolution combined with the convoluted structure of the cortex and random noise caused by the imaging method are common problems. Typically for medical images, the volumes are three-dimensional (3D), which complicates the task even further compared to two-dimensional (2D) images. Overcoming these difficulties requires task-specific algorithms. (See Liew et al. [32]).

As segmentation only defines pixels (or *voxels* in 3D) belonging to a particular segment or class, *reconstruction* is a step beyond this. Cortical reconstruction is defined to be the localization and representation of the cortical surfaces. An accurate reconstruction must correspond to the actual geometry of the cortex. Cortical segmentation and reconstruction algorithms have attracted plenty of attention recently, and surveys on the subject are available. Articles by Suri et al. [51,52], Clarke et al. [11] and Liew et al. [32] give a good overview on different approaches to the problem.

In this work, a software was developed that provides an automatic, accurate, robust and computationally efficient reconstruction of the cortex. The software is based on a previously published method that was further developed by improving approaches that are considered weaknesses. The results were evaluated with a landmark accuracy study using a substantially larger data set than in previously published studies. The software was implemented so that it completes the reconstruction task on a normal desktop computer with reasonable computational expense.

The work is funded by the TEKES project "Statistical atlas in medical image processing applications" (ATLAS) and motivated by the needs of Elekta Neuromag Oy, for which the cortex reconstruction will provide invaluable help in the inverse problem, as presented above. In addition, the software will be used in a study at Helsinki University of Technology for Alzheimer disease diagnosis and assessment, and in other future brain research projects as well. Furthermore, the software is designed and written so that with minor modification it is no longer restricted to only brain MR images, and a similar approach will be tested on other image modalities as well.

After this introduction, the second chapter presents some background to the topic and describes different approaches to cortical segmentation and reconstruction. The third chapter describes the details of the chosen reconstruction method and its numerical implementation.

The evaluation results are documented in the fourth chapter and the fifth chapter concludes the work with a discussion.

Chapter 2

Background

2.1 Magnetic Resonance Imaging

MRI is an imaging technique used especially for medical purposes. It provides high-quality images from within the human body. MRI is well-suited for brain imaging, since it is noninvasive, it produces images with high spatial resolution, and the contrast is excellent for images of soft tissues. (See Jiménez-Alaniz et al. [28]). In this section, the principle of MRI is described briefly. An example of an MRI image slice is in figure 2.1.

MRI is based on the phenomenon called nuclear magnetic resonance. The atomic nuclei have a property called *spin*, which is associated with a magnetic moment. For the purposes of medical imaging with MRI, we are interested in the hydrogen atoms inside the human body. Hydrogen atoms appear especially in water and fat tissue, and the nucleus of the most common naturally occurring isotope of hydrogen consists of a single proton which has a large magnetic moment. When placed in a strong magnetic field, the spins line up slightly more parallel than antiparallel to the magnetic field, which produces a net magnetization in the direction of the magnetic field. (See Dowsett et al. [17]).

After placing the subject in the magnetic field, a radio-frequency (RF) pulse with frequency specific to the hydrogen nucleus is applied at the target of examination. The pulse causes the atomic nuclei with the same frequency to absorb electromagnetic energy and disorientates the spins from their alignment in the magnetic field. This is called *excitation*. After the RF pulse terminates, the spins begin to return to their previous alignment. This is called *relaxation*. In relaxation, electromagnetic energy is released and detected as a signal in the receiver coil. Relaxation combines two mechanisms, the longitudinal and transverse relaxation, which correspond to longitudinal magnetization recovery and transverse magnetization decay, respectively. The longitudinal magnetization recovery rate is characterized by a tissue-specific time constant T_1, and in a similar way, transverse magnetization decay rate is characterized by a tissue-specific time constant T_2. The signal originating in relaxation depends on T_1, T_2 and proton density (PD) of the tissue. (See Dowsett et al. [17]).

To obtain contrast between different tissues, the pulse sequence and signal acquisition pa-

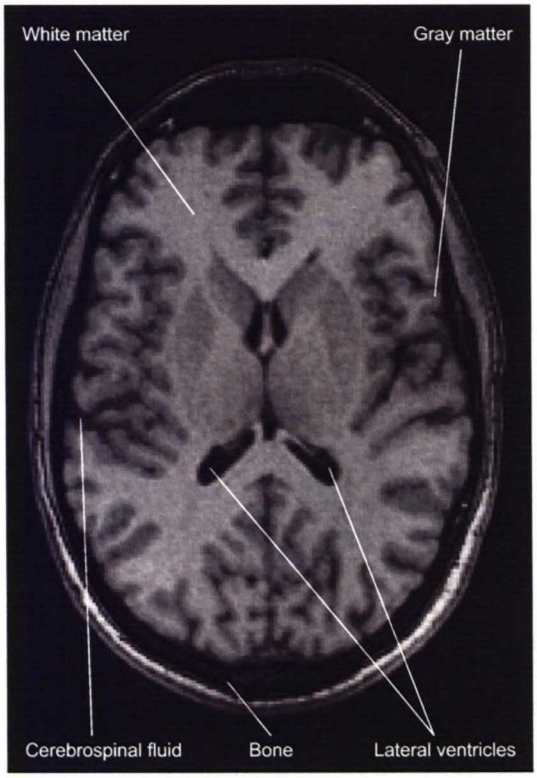


Figure 2.1 *An axial slice of a brain MR image*

rameters are chosen according to the application. This way, the effect of T_1, T_2 and PD of the tissue on the resulting signal can be adjusted, generating T_1-, T_2- or PD-weighted images. For example, T_1-weighted images are commonly used for neuroanatomical analysis, since they provide the best contrast between gray matter (GM) and white matter (WM). (See Zeng et al. [64]).

To know the exact place where a signal originated from, the coordinates in the 3D space need to be known. One of the coordinates is known by selecting a two-dimensional slice of the target at a time. The other two coordinates can be computed from the phase and frequency components of the collected signal. The final image slice is obtained by applying an inverse Fourier transform on the captured signal. The resulting 3D volume is constructed by combining the 2D slice images. (See Rajapakse et al. [43], Liew et al. [32])

It is also possible to produce more than one image from a certain position with different parameters of the imaging device. This produces multi-channel data, which can be useful for capturing data that is not present in a single-channel image. Most of the segmentation and reconstruction methods covered in this work are designed for single-channel images, some of them including an extension for multi-channel images. All of the algorithms are applicable for single-channel images. (See Zeng et al. [64]).

2.2 Structure of the Brain

The human central nervous system consists of the brain and the spinal cord, immersed in the cerebrospinal fluid (CSF). The brain consists of four principal parts: brain stem, cerebellum, diencephalon and cerebrum. The brain stem is continuous with the spinal cord and consists of the medulla oblongata, pons, and midbrain. The cerebellum is located posterior to the brain stem. The diencephalon includes the thalamus, hypothalamus, epithalamus and the subthalamus and is located superior to the brain stem. Most of the cranium is occupied by the cerebrum, which spreads over the diencephalon. The cerebrum includes the limbic system, basal ganglia, olfactory bulb, and cerebral cortex. (See Eskildsen et al. [18], Han et al. [24], Tortora et al. [54]).

The cerebrum is divided into two hemispheres, and the outer layer is known as the cerebral cortex. The cerebral cortex is a thin and convoluted layer of gray matter, which consists of closely packed neuron cell bodies. Different parts of the cerebral cortex are involved with muscle control, sensory perceptions, memory, emotions and speech (See Tortora et al. [54]). Depending on the source, the cortex is estimated to be 1–5 mm, 2–3 mm or 2–4 mm thick, and the average thickness is about 2.5 mm (See Han et al. [24], Suri et al. [51], Tortora et al. [54]).

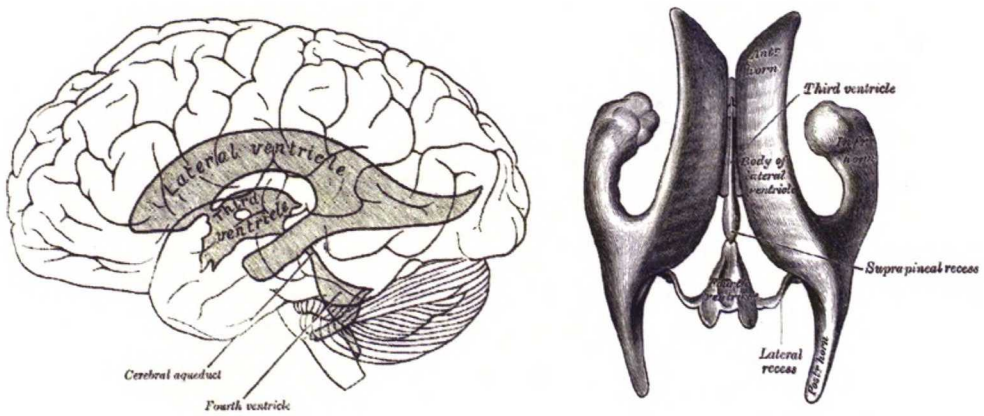
The cerebral cortex encloses the cerebral white matter. White matter consists mainly of long, myelinated axons, which transmit nerve impulses between different parts of the brain and between the brain and the spinal cord. (See Tortora et al. [54]).

Within the brain, there are cavities called *ventricles* filled with cerebrospinal fluid, depicted in figure 2.2. The largest two of the total four ventricles are the left and right *lateral ventricles* located in the cerebral hemispheres. The *third ventricle* is bounded by the thalamus and hypothalamus and the *fourth ventricle* is located lowest. The ventricles are connected to each other and they are continuous with the central canal of the spinal cord. (See Tortora et al. [54]).

2.3 Image Segmentation

Segmentation of images traditionally means dividing the image into parts or *segments* that have the following features:

1. The segments are connected. Every point within a segment is connected with a continuous path to all the other points in the same segment.
2. The segments are homogeneous with respect to defined criteria. The homogeneity criteria can be intensity or texture, for example.
3. The segments do not overlap.



(a) (b)

Figure 2.2 *(a) Relations of the ventricles to the surface of the brain and (b) cast of the ventricular cavities, viewed from above. Figures from Gray's Anatomy of the Human Body, 20th edition, 1918.*

Using a formal expression, if the whole image is I, the segmentation task is to define the K connected segments S_i so that

$$I = \bigcup_{i=1}^{K} S_i \tag{2.1}$$

where $S_i \cap S_j = \emptyset$ when $i \neq j$. In the case of medical images, the segmentation ideally finds segments that correspond to real anatomical structures or areas of interest. (See Pham et al. [39]).

Often it is suitable to ignore the connectedness rule of segmentation. In this case the term segmentation is strictly speaking not valid anymore and the operation is called pixel classification (or *voxel* classification in 3D images). The groups that the image is then divided into, are denoted as *classes*. Considering medical images, pixel classification is often more applicable than traditional segmentation, because usually points originating from the same tissue type are searched, regardless of their position. (See Pham et al. [39]).

In pixel classification it can be hard to define the number of classes a priori in some applications. Medical images have the advantage that the possible tissue types appearing in an image are often known beforehand, and this information can be used to decide the number of classes used in the classification. (See Pham et al. [39]). In this work we discuss both segmentation and pixel classification methods. For simplicity, the term *segmentation* is used to refer to both of them and the term *class* is used to refer to the resulting sets of points.

In the result of traditional segmentation one image point can belong to just one class. Sometimes this condition is too restricting, and a value of membership for different classes is used. For each voxel, the result then tells "how much" it belongs to each class. This retains more information of uncertainty from the original image. This kind of segmentation is called *fuzzy* or *soft segmentation*, compared to the traditional *hard segmentation*. The result of fuzzy segmentation is naturally easy to modify into a hard segmentation result, and usually

fuzzy segmentation is used merely as an intermediate stage of the process. More information on the subject is available in articles by Pham et al. [39] and Xu et al. [63].

2.4 Cortical Segmentation and Reconstruction

2.4.1 Goal and Challenges

The segmentation methods for brain MR images usually try to achieve either one of the following two goals: 1) distinguishing between cerebrospinal fluid, gray matter and white matter, or 2) detecting abnormalities in the brain volume. Here we focus on algorithms designed for the first goal, and in particular segmenting and reconstructing the cerebral cortex. Even though several algorithms are explicitly designed for segmentation of the cortex, also more general algorithms for segmenting the image in different tissue classes are used for the task. (See Suri et al. [51]).

Cortical reconstruction methods are designed to find cortical surfaces that enclose the cerebral cortex. This means finding the transition between white and gray matter, and the transition between gray matter and cerebrospinal fluid. Some methods also attempt to find the geometric middle of the cortex.

Typical challenges faced in segmentation and reconstruction of MR images are (see Suri et al. [51]):

1. Intensity inhomogeneity caused by non-uniformity of brain tissues and magnetic field of the imaging device
2. Partial volume effect which appears when more than one tissue contribute to the intensity of one voxel
3. Variability between subjects and the convoluted structure of the cortex, which in combination with the partial volume effect causes details to disappear
4. Random noise associated with the imaging device
5. Computational efficiency, made especially important by three-dimensional, typically large data volumes
6. Inconsistency of data between different sources
7. The reconstructed cortical surface must correspond to the human anatomy by being smooth, devoid of self-intersections and it must have the correct spherical topology (ignoring the small opening near the brain stem)

Inconsistency of data is caused by the fact that the intensity scale in MR images has no physical meaning. The image values are determined from the pulse sequence, and they are dependent on the used scanner and post-processing parameters. This causes variety in the resulting images and makes it important for a segmentation algorithm to adapt to new data sets from different sources. (See Cocosco et al. [13]).

The image volumes are 3D, which complicates the segmentation task compared to 2D images. Some algorithms use independent 2D image slices for the segmentation to achieve efficiency, but this usually lowers the accuracy of the segmentation as the 3D neighborhood of voxels is ignored.

Several algorithms have specifically been developed for correcting the intensity inhomogeneity in MR images. Many segmentation methods use one of these methods in the preprocessing stage before the actual segmentation. These correction algorithms are discussed in a recent survey by Belaroussi et al. [7]. Another important step before cortical segmentation is skull-stripping, which means removing parts from the image that correspond to tissues outside the CSF, such as the eyes, skull and skin. This step is also assumed to be taken beforehand by most brain tissue and cortex segmentation algorithms. Several skull-stripping methods are mentioned for instance in the work of Shattuck et al. [49].

2.4.2 Approaches

Numerous methods have been developed for cortical segmentation and reconstruction, and surveys on the subject are available. The most extensive survey by Suri et al. [51, 52] published in 2002 divides the methods in three categories: 1) region-based, 2) boundary/surface-based and 3) methods that fuse region-based and boundary/surface-based approaches. The survey concludes that the best results are typically given by fusion methods. A more recent survey by Liew et al. [32] includes some newer algorithms, and it also divides the methods in three similar categories: classifier-based, region-based and contour-based methods. In one of the earliest and most popular surveys from 1995, Clarke et al. [11] chose a different division, but still in three categories: threshold-based, statistical and region growing-methods.

Taking into account recent research in cortical segmentation and reconstruction methods, two directions seem to prevail: region-based methods and deformable models. The region-based methods can be divided in three distinctive subcategories: statistical, clustering, and hierarchical methods, whereas deformable models include two different approaches, the parametric deformable model and geometric deformable model. Many methods include steps from different categories, so they could also be considered as fusion algorithms rather than strictly belonging into a specific category.

Region-based methods correspond to segmentation rather than reconstruction. These methods attempt to find consistent regions for all the different tissue classes and label them accordingly. Region-based methods do not typically consider topological constraints of structures. They often suffer from misclassification of voxels, especially in data where the partial volume effect is present. This makes it hard to achieve consistency of regions in the final representation. (See Suri et al. [51]). They also require further processing to achieve coherent representations of brain structures or measures other than volume (see Zeng et al. [64]).

Currently popular approaches to segmentation and reconstruction are methods based on deformable models. Deformable models correspond more to cortex reconstruction than segmentation by tissue classification. They are designed to move contours or surfaces to the location of the cortex boundaries, separating it from CSF on the outside and WM on the

inside. In the 3D case they are called *deformable surfaces*, analog to the 2D *active contours*, also known as *snakes* (see Kass et al. [29]). In 3D also the name *active surface* can be used. Compared to region-based methods, deformable surfaces offer the benefit of getting a smooth representation of the cortex and structural information straight from the results. This means that evaluation of features such as surface area, curvature and thickness is possible. Structural information can often be used already in designing the deformable model, for example restrictions on cortex thickness, smoothness and convolutedness. This way the result will correspond more to the actual anatomy of the cortex. The resulting surface can be designed so that it retains the spherical topology of the original template. However, finding tight and deep cortical folds is still difficult for traditional deformable models. This problem is said to be overcome to some extent in some methods by using white matter instead of gray matter as the target of the surface, and constructing the inner surface of the cortex first. This information can then be used in constructing the outer surface. (See Dale et al. [15], Xu et al. [63], Zeng et al. [64]).

Since typically it is not appropriate to apply a deformable model straight on image intensity data, several successful cortex reconstruction methods combine a region-based approach with a deformable model. This way the region-based method provides a rough segmentation of the tissue classes and the deformable model acts as a refinement stage to obtain a smooth and topologically correct surface from the classification. Great benefits can be obtained by using a suitable combination of methods.

Some methods use a statistical atlas based on anatomical knowledge as prior information for the position and structure of different brain tissues. Each voxel of the image is given a label corresponding to the most likely tissue class when comparing it to the atlas. This labeling or *registration* of the image to the atlas is very effective in recognizing subcortical structures, but typically not accurate enough for segmentation of the highly convoluted cortex. For this reason in segmentation it is often used only as an initial step, and the registration result is used as a prior probability of voxels belonging to different tissue types. Registration is typically computationally expensive and it requires the whole three-dimensional image. In addition, the registration may end up with an inaccurate result and cause an undesired segmentation result if the registered image represents an abnormal, damaged or diseased brain. Registration is also dependent on the availability and accuracy of the atlas. (See Greenspan et al. [22], Shattuck et al. [49], Baillard et al. [3], Awate et al. [2]).

2.4.3 Region-Based Methods

Statistical methods in their basic form consider the image intensity values as independent samples drawn from a population. According to the theory, different classes of the population can then be modeled by probability density functions. Statistical methods are based on the gray level characteristics of images, and incorporating other features can be difficult, in particular topological constraints (see Dokládal et al. [16]). Statistical methods are typically computationally expensive (see Suri et al. [51]).

Basic statistical methods do not consider the spatial correlation of the samples, and therefore they lose in classification accuracy when considering images, especially with noise present.

For this reason adding spatial knowledge in the process with Markov Random Field (MRF) models can be very helpful (see Zhang et al. [65]). A MRF model provides a way of incorporating prior knowledge of spatial correlations to the segmentation process and thus can be used to reduce outliers and spurious classifications caused by noise. A MRF model specifies local characteristics of an image using a conditional probability model. The downsides of MRF models are said to be difficulty of implementation and computational expensiveness. (See Held et al. [26], Ibrahim et al. [27]).

In most of the statistical methods using the MRF, the segmentation is done with a statistical maximum a posteriori (MAP) classifier using three tissue classes (see Held et al. [26], Awate et al. [2]) or more when using additional transition classes (see Ruan et al. [44], Shattuck et al. [49]). MAP uses an estimator to be maximized, which is defined with the Bayes rule

$$p(\boldsymbol{X}|\boldsymbol{Z}) \propto p(\boldsymbol{Z}|\boldsymbol{X})p(\boldsymbol{X}) \tag{2.2}$$

where $\boldsymbol{X}$ is a vector representing the segmentation, $\boldsymbol{Z}$ is the image signal, $p(\boldsymbol{X})$ is a probability function incorporating a prior MRF model for the segmentation and $p(\boldsymbol{Z}|\boldsymbol{X})$ is the conditional probability of the image signal for a given segmentation.

Instead of the MAP estimator, also the expectation maximization method is used in combination with an MRF model, as in the work of Van Leemput et al. [58]. Other statistical methods have recently been published by Zhang et al. [65], Marroquin et al. [35] and Rajapakse et al. [43].

Clustering methods use unsupervised clustering at the core of the algorithm. Clustering is usually relatively easy to implement and its computational requirements are typically low, but clustering methods require some kind of handling of spatial correlation to achieve plausible results.

One popular clustering algorithm for MR images is the fuzzy c-means (FCM) clustering, introduced originally by Bezdek [9] and explained closer later on in this work. FCM or a modification of it has recently been used for cortical segmentation by Kong et al. [30], Bazin et al. [6], Xu et al. [63] and Han et al. [24]. Another commonly used clustering method is the k-means clustering (see Theodoridis et al. [53]), which is closely related to FCM. k-means clustering has been recently used in cortical segmentation by Cocosco et al. [13]. In addition, the method of Jiménez-Alaniz et al. [28] used clustering in combination with a MAP classifier, where clusters were combined based on the classification.

Hierarchical methods use the spatial information of the image directly. The hierarchical approach can be bottom-up or top-down. Bottom-up approaches start with very small segments and combine them to form larger ones. Top-down approaches start with a single or a few segments, and the segments with nonuniform samples are recursively divided. Hierarchical methods offer easy implementation and relatively inexpensive computation. Hierarchical methods do not always need additional spatial information, which differs from other region-based methods.

Wang et al. [60] proposed recently a bottom-up approach where histogram analysis was used to define which segments were combined. A top-down method introduced by Greenspan et

al. [22] starts by modeling segments with 4-dimensional Gaussian functions that take spatial correlations into account. The method is initialized with only a few segments, and the Gaussian function associated with a segment is compared to the intensity values of the voxels within the segment. Segments that include unexpected values according to this comparison are then split recursively.

2.4.4 Deformable Models

Parametric deformable models are based on the *Lagrangian* formulation of the interface evolution equation. Suppose that a velocity field $\vec{V} = [u, v, w]$ is known for each point $\mathbf{x}$ on the moving front and given as $\vec{V}(\mathbf{x})$. The movement of the front is then found by solving the ordinary differential equation

$$\frac{\partial \mathbf{x}}{\partial t} = \vec{V}(\mathbf{x}) \tag{2.3}$$

for every point $\mathbf{x}$. Since there are an infinite number of points on the front, it has to be discretized. This means using line segments in 2D or triangles in three dimensions, and moving the end points of these elements accordingly. (See Osher et al. [38]).

The contour or surface formed by the discretized points moves to an energy-minimizing position with respect to a predefined energy function. The energy function consists of internal forces that affect the shape of the contour, external forces that are determined from the image and possibly external constraints that can be defined according to the requirements of the application. The external forces are chosen so that the contour settles in a position that corresponds to image boundaries of interest. (See Xu et al. [61], Goldenberg et al. [20], Segonne et al. [46]).

The classic "snake" model was introduced by Kass et al. [29] where the contour is represented by $C(q)$ and the total energy $E(C)$ defined as

$$E(C) = \alpha \int |C'(q)|^2 dq + \beta \int |C''(q)|^2 dq - \lambda \int |\nabla I(C(q))| dq \tag{2.4}$$

where α, β and λ are positive constants and ∇I is the spatial gradient of the image. The first two terms correspond to the internal energy controlling the smoothness of the contour and the third term attracts the contour towards the object of interest in the image. (See Caselles et al. [10]).

Methods based on parametric active contours have some difficulties in common. The model is sensitive to the parameters of the force model and the initial position of the contour. The discretization of boundary elements has to be periodically modified to prevent distortion of the evolving contour. Self-intersections of the surface are computationally expensive to prevent, and topological changes are difficult to handle. Parametric deformable models may also experience difficulties with significant protrusions. (See Osher et al. [38], Baillard et al. [3], Awate et al. [2]). Parametric deformable models have recently been used for cortical reconstruction by Xu et al. [63], Eskildsen et al. [18], MacDonald et al. [33] and Dale et al. [15].

Geometric deformable models were first introduced by Caselles et al. [10] and Malladi et al. [34], where propagating interfaces are represented implicitly as level sets of higher-dimensional, scalar level set functions (see Pham et al. [23]). Geometric deformable models evolve in an *Eulerian* fashion, since the interface is *captured* by the level set function, as opposed to being tracked by interface elements as in the Lagrangian formulation. (See Osher et al. [38]).

The level set theory used by the geometric deformable models was introduced by Osher and Sethian in 1988 [37] for tracking moving interfaces in a wide variety of problems. The fundamental evolution equation is defined as

$$\phi_t(\mathbf{x},t) + \vec{V}(\mathbf{x},t) \cdot \nabla\phi(\mathbf{x},t) = 0 \tag{2.5}$$

where the first term represents the partial derivative of the implicit level set function $\phi(\mathbf{x},t)$ with respect to its time variable t, $\vec{V}(\mathbf{x},t)$ is the velocity at each point $\mathbf{x}$ on the implicit surface at a certain point of time and ∇ is the gradient operator

$$\nabla\phi(\mathbf{x},t) = (\phi_{x_1}(\mathbf{x},t), \phi_{x_2}(\mathbf{x},t), \phi_{x_3}(\mathbf{x},t)) \tag{2.6}$$

Equation (2.5) is often called the *level set equation.* The velocity $\vec{V}(\mathbf{x},t)$ can depend on different factors, such as an external velocity field or the geometry of the level set function. (See Osher et al. [38]).

Geometric deformable models have several advantages over parametric deformable models. Geometric deformable models are completely intrinsic, so no nodes have to be added or removed as in parametric models. Geometric properties of the level set function are easy to compute, such as the normal vector or local curvature. In addition, topological changes require no special handling and self-intersections of the contour are not possible. (See Han et al. [23]).

Topological flexibility of the geometric deformable models is usually considered as a great advantage but it also means that topological changes are difficult to prevent. This can be a significant problem when the initial topology of the surface is correct and preserving it is important (see Han et al. [23]).

Geometric deformable models have recently been used for cortical reconstruction by Li et al. [31], Zeng et al. [64], Goldenberg et al. [20] and Han et al. [24]. The latter continued and improved from the work of Xu et al. [63], which was assessed to be one of the best-performing algorithms in the survey by Suri et al. [51,52]. The method proposed by Han et al. [24] was chosen for closer study and implementation in this work.

2.4.5 Segmentation Evaluation

The evaluation of cortex segmentation results is difficult. Generally for segmentation, the evaluation is preferably done by comparing the results produced by the method to the "true" segmentation. Naturally, no ground truth is available for in vivo MR images. Method evaluation is commonly done by either comparing the result to a manually segmented image,

or alternatively using synthetic brain phantoms which have the information of the underlying segmentation attached to them. (See Greenspan et al. [22], Cuadra et al. [14]).

Using real images for evaluation is advisable to achieve plausible results. However, generating manual segmentations is very laborious, and in addition the quality may fluctuate because of the tremendous amount of data and variability of manual segmentation results depending on the operator. These factors limit the availability of reliable data. Many of the recently published methods are validated with manually segmented real MR brain images offered by the Internet Brain Segmentation Repository (IBSR), available online at http://www.cma.mgh.harvard.edu/ibsr/.

Synthetic images aim to model the features of real MR images. Simulated images are popular in method evaluation, since they offer unlimited data and a precise error estimate (see Cuadra et al. [14]). Synthetic images can be generated for instance using a free simulator offered by BrainWeb [12], which has been used in several studies such as ones by Greenspan et al. [22], Ibrahim et al. [27], Jiménez-Alaniz et al. [28] and Awate et al. [2].

One other approach to evaluation is to compare the result to a result by some other segmentation method. The fuzzy c-means clustering is often used as a comparison for the accuracy of region-based algorithms. The IBSR website also provides segmentations by some basic methods.

Chapter 3

Methods

3.1 Outline

This chapter is a detailed presentation of the cortex reconstruction method implemented in this work. It is a slightly modified version of a method originally proposed by Han et al. [24], called Cortical Reconstruction using Implicit Surface Evolution (CRUISE). This implementation uses roughly the same overall structure, but some stages were performed in a different fashion. The ventricle filling stage of CRUISE was replaced with a significantly simpler approach, reducing computational expense and removing the need to register the image to a particular coordinate system. In addition, in this work the GM membership editing stage and gradient vector flow computation were performed based on the reconstruction of the inner cortical surface, rather than straight from the clustering results as in CRUISE. This was considered a reasonable modification, since the reconstruction result is assumed to give higher accuracy than the clustering, and consequently resulting to better results for both the central and outer surfaces.

CRUISE was chosen for implementation for its reported accuracy, robustness and the fact that it reconstructs also the central surface, whereas most methods only reconstruct the inner and outer surfaces. It seemingly represents the current state-of-the-art.

The method takes raw brain MR data as input and generates three surfaces as output, corresponding to 1) the inner cortical surface, which is the transition point between WM and GM, 2) the outer cortical surface, which is the transition between GM and CSF and 3) the central cortical surface, which is at the geometric center between the inner and outer surfaces. The topology of the surfaces corresponds to human anatomy, being spherical and fully connected without handles or holes.

Different stages of the method are briefly described here, a more detailed discussion is provided in the following sections. Figure 3.1 shows the overall structure of the reconstruction method.

The method starts with T1-weighted volumetric MR brain images as input. Preprocessing (stage 1 in figure 3.1) consists of three steps: brain extraction, cerebellum removal and

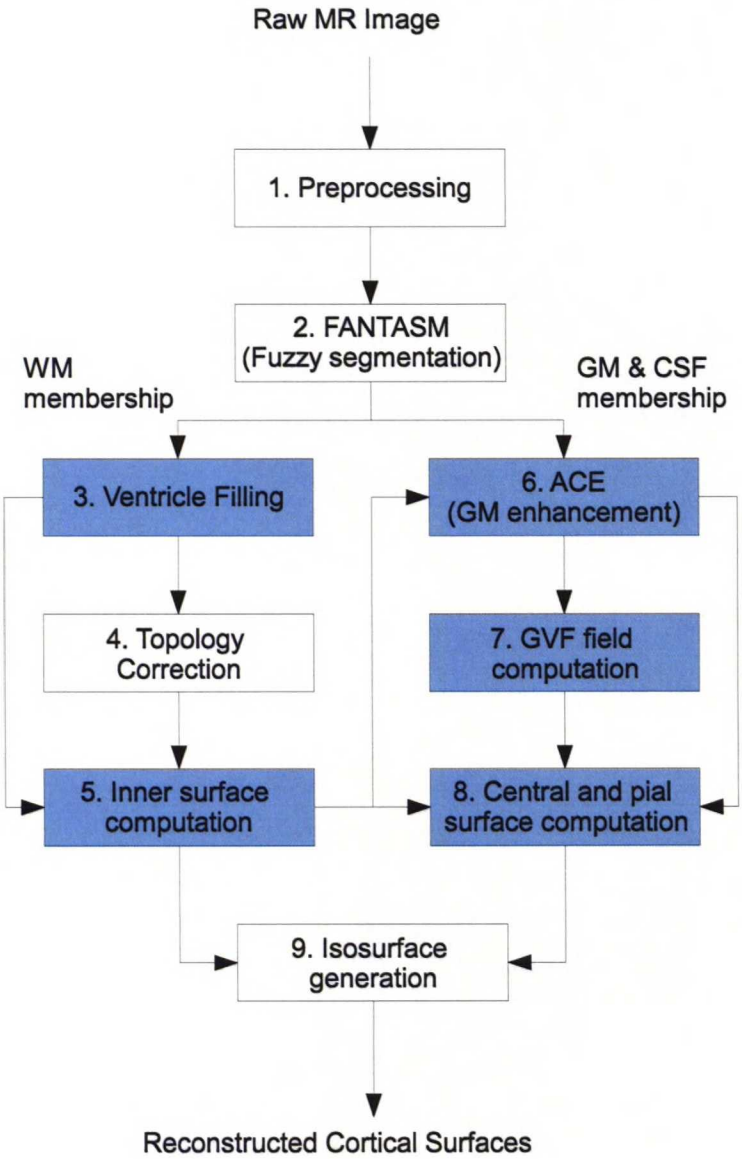


Figure 3.1 *Overall structure of the reconstruction method. Stages marked with blue boxes were implemented in this work, of which stages numbered 5 to 8 are integrated in a single software. The remaining stages were performed with publicly available tools.*

reslicing of the volume to obtain isotropic voxels. After preprocessing only three tissue types are left in the image volume and fuzzy clustering (stage 2) is used to classify voxels in three classes. The subsequent clustering results or *membership functions* hold information of the uncertainty of a voxel belonging to a certain class, and this is valuable at a later stage of the method (see Pham [42]). A binarized version of the membership function corresponding to white matter is also generated.

Parts corresponding to the ventricular system are filled in both the binarized and original WM membership function (stage 3). Topology correction (stage 4) is performed on the binarized and filled WM membership volume, resulting in an approximation of the inner cortical

surface, with a spherical topology corresponding to human anatomy. This approximation is used as initialization for the first geometric deformable model. The geometric deformable model evolves to a steady state as determined by the original WM membership function, resulting in a distance map corresponding to the inner cortical surface (stage 5).

The inner cortical surface computation result is used for finding and opening cortical folds in the GM membership function, by decreasing the GM membership value at points interpreted as cortical folds (stage 6). After this, follows the computation of a gradient vector flow (GVF) field that generates vectors pointing to the center of the edited GM membership function (stage 7). The central surface is computed by a geometric deformable model initialized with the inner surface computation result and directed by the GVF field, and subsequently the result is used to initialize another geometric deformable model for computing the outer cortical surface (stage 8). The geometric deformable model for the outer cortical surface is directed by the edited fuzzy clustering results in a similar way as for the inner surface. Finally three isosurfaces are generated from the surface representations, creating the final reconstructions (stage 9).

The main part of the implementation in this work consists of a software that takes the fuzzy clustering results as input, together with the topology corrected initialization for the geometric deformable model. It produces the three surface representations as output, retaining the initial topology. The software was done with a modular structure so that the integration of other stages is straightforward at a later time. It is written in C programming language and performs fully automatically, and is computationally efficient. In addition, a ventricle filling method was developed and implemented, but not included in the software at this point since the following topology correction stage was conducted with an external tool.

Preprocessing, fuzzy segmentation, topology correction and some manual interaction in the ventricle filling stage were performed with MIPAV (see McAuliffe et al. [36]) and publicly available plug-ins installed for the software. BrainSuite 2.0 (see Shattuck et al. [50]) was used for cerebrum extraction in preprocessing. Preprocessing and ventricle filling stages require some manual interaction in this implementation, otherwise the method is fully automated. The isosurface generation and evaluation of results was done in MATLAB 7.0.

3.2 Preprocessing and Fuzzy Segmentation

The method starts with T1-weighted volumetric MR image data. The first step in preprocessing is to perform "skull-stripping" to remove noncerebral tissues, such as skin, bone and fat. This is done with *Brainstrip* (see Goldszal et al. [21]), a semiautomatic plug-in for MIPAV. The plug-in was developed by Christos Davatzikos and Jerry Miller at the Johns Hopkins University. Brainstrip uses thresholding, morphological operations and region growing to obtain the final result. Iteration of the method parameters is necessary in most cases before a feasible result. Stages of skull-stripping are shown in figure 3.2.

After skull-stripping, the image volume is imported in BrainSuite 2.0 [50] to extract the cerebrum from the remaining volume. The program generates automatically a mask that removes the cerebellum and parts of the brain stem by warping a template on the image

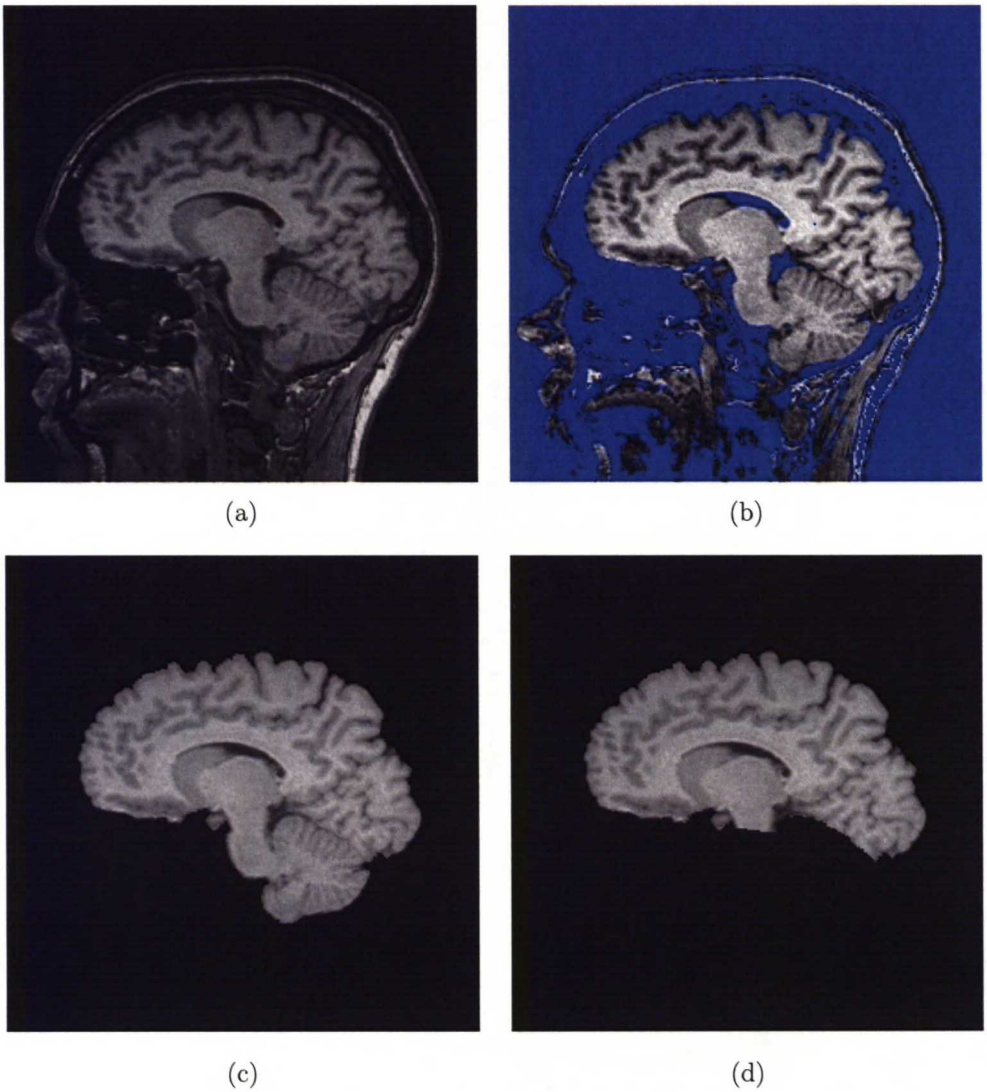


Figure 3.2 *Preprocessing steps: (a) Original image, (b) thresholding with Brainstrip tool, (c) Brainstrip result and (d) Brainsuite 2.0 cerebrum extraction result*

volume (see figure 3.2(d)). The automatically generated mask may also be modified manually before applying it on the image volume. After masking, the image was imported back to MIPAV, and as the final step of preprocessing the image volume was resliced to obtain isotropic voxels using cubic b-spline interpolation.

After preprocessing, fuzzy segmentation is applied to identify the spatial distribution of WM, GM and CSF within the image volume. The method is a modification of the widely used FCM clustering. FCM clusters data by computing a value of membership at each data point for a specified number of classes. The fuzzy membership function reflects the degree of similarity between the data value at that location and the centroid of its class. The membership values are constrained to be between zero and one, with the highest values representing values closest to the centroid of that particular class. Formally FCM minimizes the following objective function with respect to the membership functions u and the centroids

v (see Pham et al. [40]):

$$J_{\mathrm{FCM}} = \sum_{j\in\Omega}\sum_{k=1}^{C} u_{jk}^{q}||y_j - v_k||^2 \tag{3.1}$$

where Ω is the set of voxel locations in the image volume, q is a parameter that controls the fuzziness of the classification and is constrained to be greater than one (if $q = 1$, FCM is equal to k-means clustering), u_{jk} is the membership value at voxel location j for class k so that $\sum_{k=1}^{C} u_{jk} = 1$, y_j is the image intensity at location j, and v_k is the centroid of class k. The objective function is minimized when high values are assigned to voxels with intensities close to the centroid of the particular class, and low values are assigned to voxels with intensities far from the centroid.

Standard FCM does not take spatial dependencies of voxels into account, so it loses valuable information when used on data with strong noise and high spatial correlation, such as MR images. Here the fuzzy segmentation is performed using an algorithm called Fuzzy And Noise Tolerant Adaptive Segmentation Method (FANTASM) (see Pham et al. [41]), which is a modification of FCM clustering. To obtain better results than FCM on MR brain images, FANTASM incorporates a spatial smoothing term to reduce the effects of noise, and also a gain field estimation term to compensate for MR intensity inhomogeneities. Using a similar notation as above for FCM (3.1), FANTASM minimizes the cost function (see Han et al. [24], Pham et al. [40], [42])

$$\begin{aligned} J_{\mathrm{FANTASM}} = &\sum_{j\in\Omega}\sum_{k=1}^{C} u_{jk}^{q}||y_j - g_j v_k||^2 + \frac{\beta}{2}\sum_{j\in\Omega}\sum_{k=1}^{C} u_{jk}^{q}\sum_{l\in N_j}\sum_{m\neq k}^{C} u_{lm}^{q} \\ &+\lambda_1\sum_{j\in\Omega}\sum_{r=1}^{3}(D_r * g)_j^2 + \lambda_2\sum_{j\in\Omega}\sum_{r=1}^{3}\sum_{s=1}^{3}(D_r * D_s * g)_j^2 \end{aligned} \tag{3.2}$$

Here the second term performs the spatial smoothing and the last terms are first- and second-order regularization terms to ensure that the gain field (denoted by g_j) is spatially smooth and slowly varying. N_j represents the set of first order neighbors of voxel j, the constants β, λ_1 and λ_2 are weights that determine the smoothness between neighboring voxels in the resulting membership functions (β) and the strength of the gain field (λ_1 and λ_2). D_r (D_s) is a finite difference operator along the rth (sth) dimension of the image and $D * g$ refers to convolution of g with the kernel D. In this implementation, FANTASM is computed using a MIPAV plug-in by Pierre-Louis Bazin and Dzung L. Pham. The values used for the variables were $q = 2$, $C = 3$, $\beta = 0.05$, $\lambda_1 = 3$ and $\lambda_2 = 3$. The membership functions corresponding to WM, GM and CSF are later denoted by μ_{WM}, μ_{GM} and μ_{CSF}, respectively. Figure 3.3 shows an example of each of the three membership functions produced by the fuzzy clustering.

3.3 Ventricle Filling

The WM membership function μ_{WM} generated in the previous stage by fuzzy segmentation is later used to direct the geometric deformable model to find the inner cortical surface. However, a couple of steps have to be taken before that to obtain desired results. "Outside" μ_{WM} can be considered as an approximation to the inner cortical surface, with the exception of cavities inside μ_{WM} corresponding to caudate nucleus, putamen and the ventricular

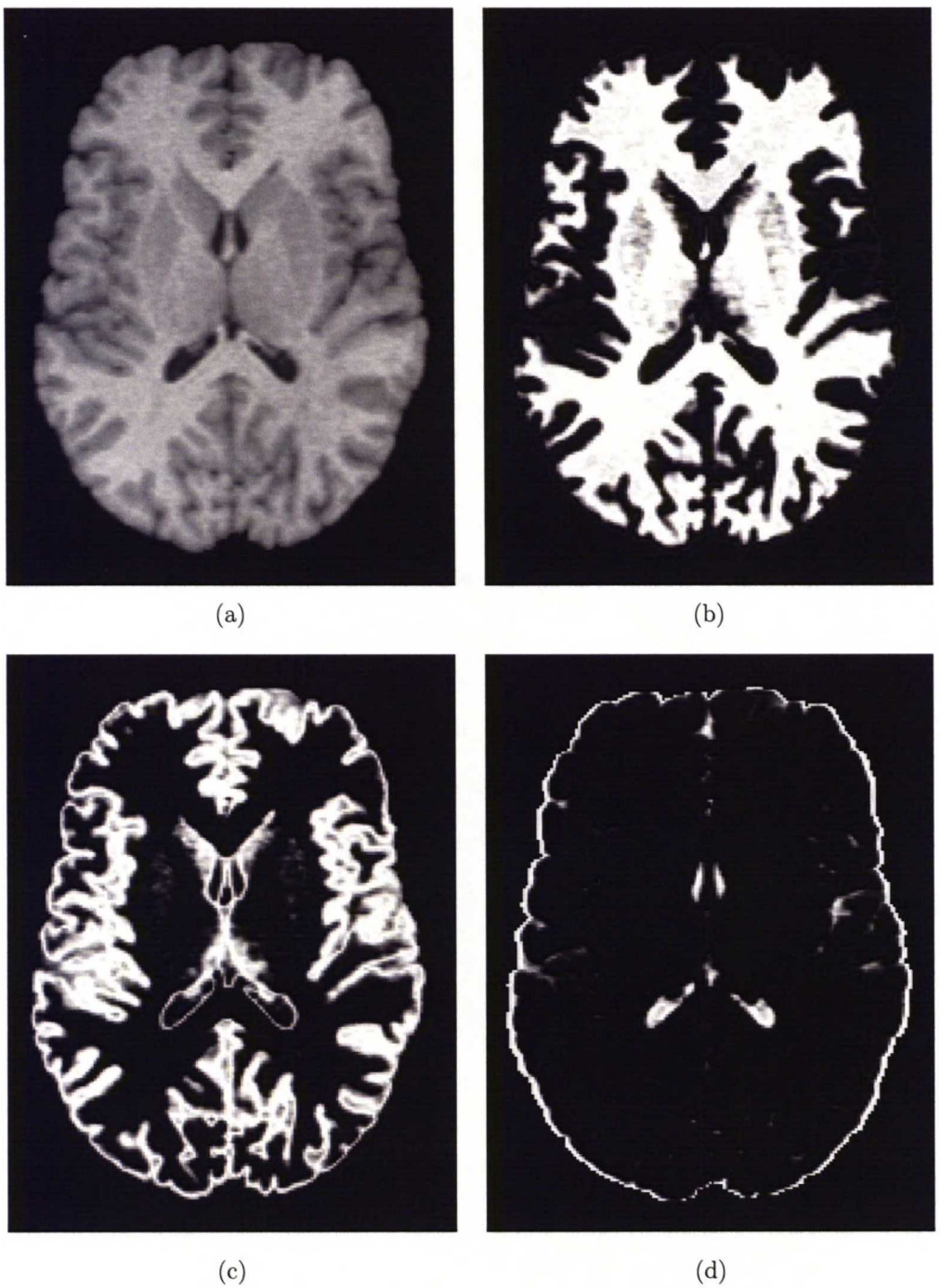


Figure 3.3 *Axial cross-section of fuzzy clustering results: (a) Original image, (b) WM membership, (c) GM membership and (d) CSF membership. It is worth noting that in (c) apart from cortical GM, also sub-cortical GM and the putamen region are assigned high membership values.*

system. In this stage, μ_{WM} is edited by filling these cavities. In addition to preventing the geometric deformable model from advancing to the ventricles at a later stage, this operation also makes topology correction of the volume possible.

Figure 3.4 *Manually drawn triangles on the binarized WM membership at locations of the middle cerebral artery. The areas inside the triangles are filled prior to the ventricle filling algorithm to prevent the background region from expanding through the openings to the center of the volume.*

In CRUISE the ventricle filling operation was done with a method known as AutoFill, proposed by Han et al. [25]. AutoFill starts by finding the approximate location of ventricles with a geometric deformable model, and then uses the result as a seed for a region-growing algorithm. The region-growing is performed in three separate passes on 2D coronal slices and bounded by various sealing lines. AutoFill also requires identifying the putamen region from the image by registration of the image to the Talaraich coordinate system.

In this work a different approach was chosen to simplify the process. The method currently requires manual interaction. It was developed specially for this purpose. The method is based on region growing and it takes advantage of the knowledge of direction where ventricles open in μ_{WM}. The resulting mask is not exactly the same shape as generated by AutoFill, but this was considered irrelevant since the goal of the step is simply to mask subcortical structures without affecting cortical surfaces.

First, μ_{WM} is thresholded at the value of 0.5 to create a binary object μ_{WMBIN}, where value of one represents the object and zero the background. Manual editing is performed on coronal slices of μ_{WMBIN} to fill small openings at locations corresponding to the middle cerebral artery (see figure 3.4). This is fairly easy to do, since the openings occur more or less at the same location on consecutive slices. A contour is drawn and copied across 8–10 slices, and the marked area is filled with the object value. The result of this manual editing is denoted by μ'_{WMBIN}.

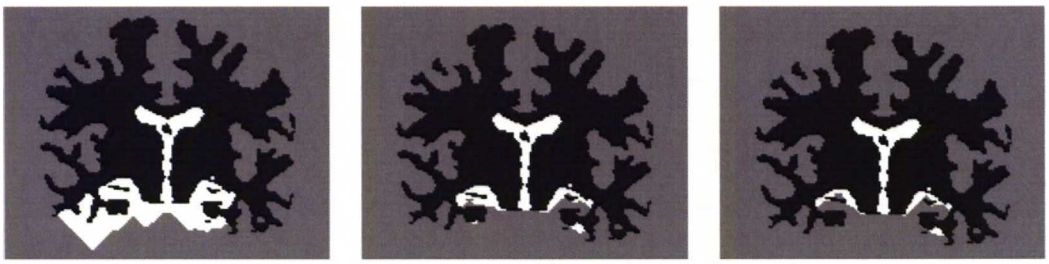
Next, a special double-front region growing is applied. Let M and M' be grids with the same dimensions as the image volume, denoted by m, n and p. The value at a grid point $\mathbf{x}$ is denoted by $M(\mathbf{x})$. The grid points $\mathbf{x} = [x_1, x_2, x_3]$ can have four different values, -1, 0, 1 or 2, which correspond to regions *white matter, undefined, concavity* and *background*, respectively. We define also an integer variable l that is used to control the advance of the regions.

Additionally, a function $F(M(\mathbf{x}))$ is used that provides information of the neighborhood around point $\mathbf{x}$. Consider a 9×9 area in the x_1, x_2-plane, centered at point $\mathbf{x_i}$. The value of function $F(M(\mathbf{x_i}))$ is equal to the number of grid points corresponding to white matter (value -1) in the defined area. This function is used to slow the advance of the concavity region in areas that are far from white matter.

The algorithm performs the following steps:

1. Initialize $M \Leftarrow -\mu'_{\mathrm{WMBIN}}$
2. Initialize $M' \Leftarrow M$ and $l \Leftarrow 0$
3. Using the coordinate axes so that x_3 decreases when advancing from top to bottom as in figure 3.6, for all points $\mathbf{x_i}$ at the top of the image volume ($x_3 = p$), assign $M(\mathbf{x_i}) \Leftarrow 2$.
4. Choose (manually) a point $\mathbf{x}_{in}$ at the top of either of the lateral ventricles, and assign $M(\mathbf{x}_{in}) \Leftarrow 1$.
5. Increase l by 1.
6. For all points $\mathbf{x_i}$ where $x_3 \geq p-l$ and $M(\mathbf{x_i}) = 0$, assign $M'(\mathbf{x_i}) \Leftarrow max([N_{26}(\mathbf{x_i}) \cap M])$. $N_{26}(\mathbf{x_i})$ represents the 26-neighborhood of point $\mathbf{x_i}$ (see section 3.6.4 for details).
7. For all points $\mathbf{x_i}$ where $M'(\mathbf{x_i}) = 1$, $x_3 = p - l$ and $F(M(\mathbf{x_i}) \geq \alpha$, assign $M'(\mathbf{x_i}) \Leftarrow 0$ (α is a non-negative integer, chosen as $\alpha = 5$).
8. If M' and M are identical, go to step 10.
9. Assign $M \Leftarrow M'$ and go to step 6.
10. If $l < p - 1$, go to step 5. Otherwise stop.

The algorithm starts at the top of the image volume, growing the two regions simultaneously within the area defined by the variable l. A pass of the region growing is done as follows: for every point $\mathbf{x_i}$ that is within the area defined by the limit ($x_3 \geq p-l$) and the mask value of the point is zero ($M(\mathbf{x_i}) = 0$), assign to the corresponding temporary mask point $M'(\mathbf{x_i})$ the highest mask value from the 26-neighborhood of the point ($M'(\mathbf{x_i}) \Leftarrow max([N_{26}(\mathbf{x_i}) \cap M])$. Next, if the assigned value was one ($M'(\mathbf{x_i}) = 1$) and the point is at the lowest slice defined by the limit ($x_3 = p - l$), the surrounding *white matter* points in the 9×9 area in x_1, x_2 plane are counted. If there are less than α points corresponding to white matter around the point $\mathbf{x_i}$, the value in the temporary mask is changed back to *undefined* ($M'(\mathbf{x_i}) \Leftarrow 0$). This way, the growing of the region inside *concavities* is slowed down at locations with only a few



(a) (b) (c)

Figure 3.5 *The effect of the variable α in slowing the concavity region growing in open areas, (a) $\alpha = 0$, (b) $\alpha = 2$, (c) $\alpha = 5$. In (c) there is still a region that has grown outside the concavities at the lower right part of the image.*

surrounding *white matter* points. This is done to prevent this region from growing to areas outside the concavities (see figure 3.5).

After this, if M' and M are not equal to each other, the temporary mask has changed in the previous pass. In this case M is set to be equal to the temporary mask M' and the steps are repeated with the same value of l. Once the regions reach a steady state in the given area, l is increased by one and the process is repeated for the new area, which is one slice larger than in the previous step. The algorithm stops when l reaches the maximum value of $p-1$. The benefit of using the limit l, is that as the ventricles open downwards, most of the area corresponding to ventricles is filled before the two regions meet when l reaches higher values (see figure 3.6).

After the region-growing has finished, the result is thresholded, creating a mask $M_t(\mathbf{x_i})$ for the concavity region

$$M_t(\mathbf{x_i}) = \begin{cases} 1, & \text{if } M(\mathbf{x_i}) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

In this version of the ventricle filling algorithm, M_t has to be often edited manually to remove regions where the generated mask has advanced to undesired parts of the image. In these cases, M_t is simply cut at the desired location and the largest connected component is chosen as the final mask.

After the mask has been generated, it is summed to both μ'_{WMBIN} and μ_{WM}, creating the filled fuzzy clustering result μ'_{WM} for directing the geometric deformable model and filled and binarized volume for topology correction μ'_{WMBINf}

$$\mu'_{\mathrm{WM}} = \mu_{\mathrm{WM}} + M_t \tag{3.4}$$

and

$$\mu'_{\mathrm{WMBINf}} = \mu'_{\mathrm{WMBIN}} + M_t \tag{3.5}$$

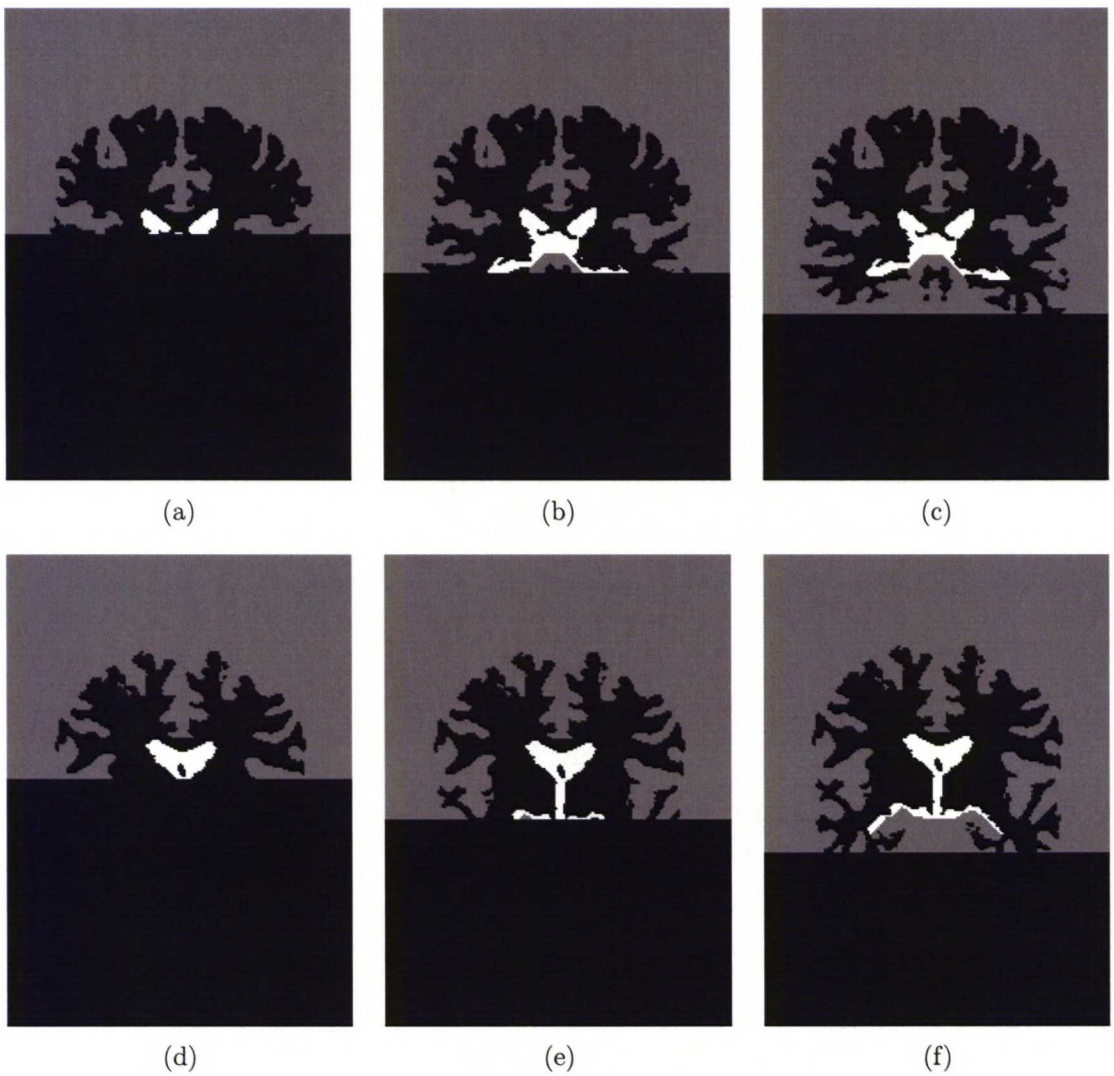
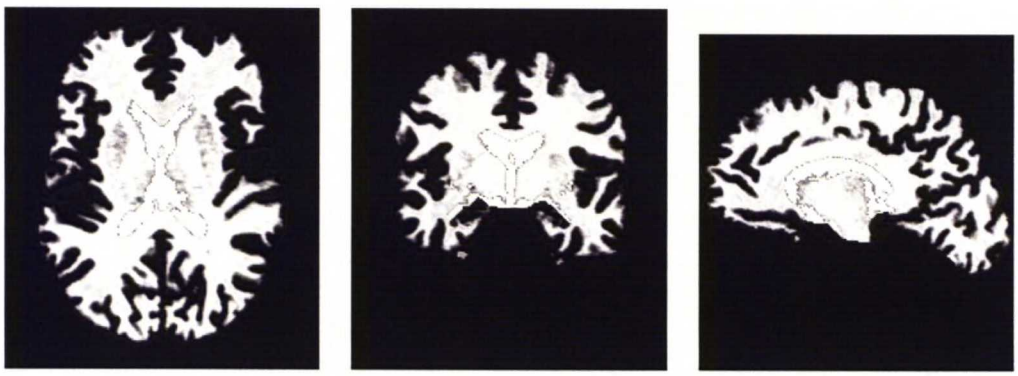


Figure 3.6 *The ventricle filling method advances in the increasing x_3-direction, shown at two cross-sections with (a),(d) $l = 116$, (b),(e) $l = 136$ and (c),(f) $l = 156$. The figure intensities are not the same values as assigned to the grid points because of choices made in the implementation. In the figures white represents the ventricle mask, gray is background and black is undefined where $x_3 > l$, and white matter otherwise.*

3.4 Topology Correction

To achieve a reconstruction result with the desired spherical topology, the geometric deformable model has to be initialized with a topologically correct surface and the topology preserved in the deformation process. The initialization is required to be close to the target object, and this is achieved by correcting the topology of the binarized and filled WM membership function μ'_{WMBINf}. The output of the topology correction is denoted by μ''_{WMBINf}.

Topology correction was done with a plug-in for MIPAV, developed by Pierre-Louis Bazin and Dzung L. Pham at Johns Hopkins University. The plug-in implements a method pro-



(a) (b) (c)

Figure 3.7 *The mask resulting from ventricle filling added to the original WM membership function, shown from three directions.*

posed also by Bazin and Pham [5], which uses the Fast Marching Method (see section 3.5.2) for the task.

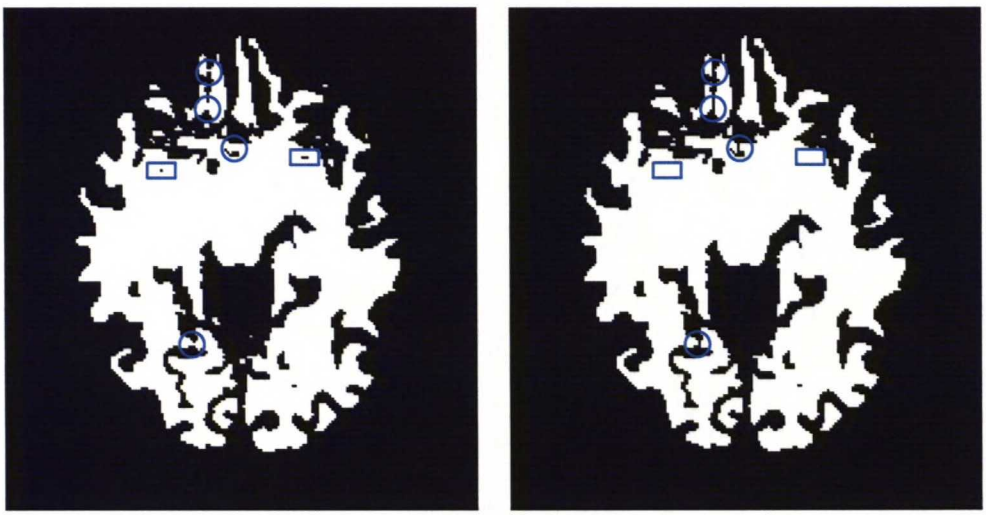
First the largest connected component is selected from μ'_{WMBINf} and other components are removed. Then all the holes within the object are removed, switching background voxel values to object values for points that have a background value but are not connected to the actual image background. Then a distance function $d(\mathbf{x})$ is computed inside μ'_{WMBINf} using the Fast Marching Method, assigning largest distance values for points furthest from the object boundary. Background points are ignored.

Let $G(t)$ be the set of points included in μ''_{WMBINf} at time step t. Beginning with an empty set as $G(-1)$, a single point with the largest distance value $d(\mathbf{x})$ is assigned as $G(0)$. This object is then grown stepwise, at each time step n adding points connected to previously accepted points $G(n-1)$ and having positive values of $d(\mathbf{x})$. In addition these points have to be *topologically simple* (see section 3.6.4). According to Bertrand in [8], a point is called a *simple point* if the topology of the object is unaltered when the point is added or removed. By adding only simple points, the initial topology is preserved. When starting with one single point as $G(0)$, this results in a spherical topology, cutting any handles appearing in μ'_{WMBINf}. Points are added in an order defined by a priority value, favoring points with the largest values of $d(\mathbf{x})$. This results in corrections appearing close to the boundary of the object. For further details, the reader is referred to the original article by Bazin et al. [5].

3.5 Anatomically Consistent Segmentation Enhancement

3.5.1 Model

A common problem faced when reconstructing the outer cortical surface is the difficulty of detecting deep and tight sulcal folds. The partial volume effect can cause information of CSF to disappear inside a cortical fold. For this reason, it is necessary to introduce a step



(a) (b)

Figure 3.8 *Binarized WM membership function (a) before and (b) after topology correction. Locations of filled holes are marked with rectangles and cut handles with circles.*

to make the folds easier to find.

This step is done with a method named *anatomically consistent GM enhancement* (ACE) (see Han et al. [24]). It is based on finding the middle of the sulcal folds and reducing the GM membership at these locations. Finding the middle of the cortical folds is done by examining shocks of fronts advancing outward from the inner cortical surface. Shocks are points where two fronts meet, and the method takes advantage of the fact that the gradient is not well-defined at a shock point. Naturally, a finite difference approximation to the gradient can still be computed, and the shock points can then be distinguished by lower gradient values in comparison to other points. Marching outward from the inner cortical surface, the opposing fronts inside a sulcal fold are expected to meet at the middle of the fold in the simplest case. The meeting point can also be adjusted, as discussed below.

The algorithm uses the Fast Marching Method, also known as the *Fast Marching Level Set Method*. It is also applied later for initializing the implicit level set function as a signed distance function. The following section 3.5.2 presents the Fast Marching Method and section 3.5.3 describes the details of the ACE algorithm.

3.5.2 The Fast Marching Level Set Method

The Fast Marching Level Set Method is an efficient algorithm for tracking the evolution of an interface that propagates normal to itself with a speed F that only depends on the position of the interface (see Sethian [47]). In addition, the speed has to be always either positive or negative. In this work, the Fast Marching Method is used first to locate the middle of sulcal folds and later for computing the signed distance function for initializing the level set function.

Consider the case of a front moving with a speed $F = F(\mathbf{x}) = F(x_1, x_2, x_3), F > 0$. We then have a monotonically advancing front with a level set equation (level set equations explained in more detail in section 3.6.1)

$$\phi_t(\mathbf{x}, t) + F(x_1, x_2, x_3)||\nabla\phi(\mathbf{x}, t)|| \tag{3.6}$$

where $\phi(\mathbf{x}, t)$ represents an implicit level set function, and the first term is its derivative for the time variable t. Let $T = T(x_1, x_2, x_3)$ be the time at which the front crosses the point $\mathbf{x} = (x_1, x_2, x_3)$. The surface T then satisfies the equation

$$||\nabla T||F = 1, \tag{3.7}$$

which expresses that the gradient of the arrival time is inversely proportional to the speed of the front. The equation is a form of the Eikonal equation. (See Sethian [47]).

Using the approximation to the gradient described in section 3.6.3, equation (3.7) can be expressed as finding the solution to

$$\begin{aligned} \frac{1}{F_{ij}^2} = &max(max(D_{ij}^{-x_1}T, 0), -min(D_{ij}^{+x_1}T, 0))^2 + \\ &max(max(D_{ij}^{-x_2}T, 0), -min(D_{ij}^{+x_2}T, 0))^2 + \\ &max(max(D_{ij}^{-x_3}T, 0), -min(D_{ij}^{+x_3}T, 0))^2 \end{aligned} \tag{3.8}$$

In the simplest case, we want to construct the signed distance function where $||\nabla T|| = 1$, so the speed is set to $F = 1$ to satisfy equation (3.7). This corresponds to computing the Euclidian distance to the initial surface. With the help of this formulation, the value for each grid point can be computed. Equation (3.8) is a quadratic equation, and the largest possible value is always selected for each grid point as the solution. In this way, the correct "viscosity solution" is obtained. (See Osher [47]).

The idea of the fast marching method is to build the solution for equation (3.8) outward, starting from the smallest value for T. An important notice is that the value of each grid point can be computed based only on the values of its neighboring points. Fast marching method takes advantage of this notice and it is initialized by dividing points in three groups:

1. Known points: The values of these points are already known.
2. Candidate points: Candidate points are adjacent to known points, and they all have a candidate value assigned. The candidate values are computed with equation (3.8), using the values of known points (group 1).
3. Far away points: All the rest of the points that do not belong to the first two groups.

The algorithm then loops the following steps:

1. Choose $(i_{min}, j_{min}, k_{min})$ as the candidate point (group 2) with the smallest candidate value.
2. Move $(i_{min}, j_{min}, k_{min})$ from candidate points (group 2) to known points (group 1).

3. If any neighbors of $(i_{min}, j_{min}, k_{min})$ are in group 3, move them to group 2. The following points are considered as neighbors: $(i_{min}-1, j_{min}, k_{min})$, $(i_{min}+1, j_{min}, k_{min})$, $(i_{min}, j_{min}-1, k_{min})$, $(i_{min}, j_{min}+1, k_{min})$, $(i_{min}, j_{min}, k_{min}-1)$, $(i_{min}, j_{min}, k_{min}+1)$.

4. Update candidate values of all neighbors of $(i_{min}, j_{min}, k_{min})$.

5. Return to step 1.

The output of Fast Marching Method are the known point values (group 1). The algorithm may be stopped at any point, so that only points up to a certain distance value are computed.

Choosing the point with the smallest candidate value (step 1) with minimal computational expense is crucial in the Fast Marching Method to achieve efficiency. To do this, the binary heap (see Sedgewick [45]) is chosen as the data structure for the set of candidate points. The binary heap is a binary tree with two additional constraints: 1) each node satisfies the *heap condition* which states that each parent node has higher priority than its children nodes and 2) the binary tree is fully filled, except possibly the last level. The binary heap is easy to represent with an array structure where the parent is in position j and its children in positions $2j$ and $2j+1$.

From the first constraint follows that the root of the heap has always the highest priority. In the Fast Marching Method, the candidate points are arranged in to a binary heap according to their candidate value, so that the point with the smallest candidate value has the highest priority. At every iteration, the root of the heap is removed and the heap is returned to satisfy the heap condition. The point corresponding to the removed root node is added to the set of known points (group 1) and removed from the candidate points (group 2).

3.5.3 Editing the Membership Function

In ACE, the speed term of the Fast Marching Method is assigned to change according to the CSF membership function. This is done to slow the front down when CSF membership has high values. In this way, the point where the two opposing fronts meet and form the shock (see section 3.5.1) is close to high values of CSF membership, rather than in the middle as defined by ordinary Euclidian distance. This modification should direct the meeting point of the fronts towards possible evidence of cerebrospinal fluid inside the cortical fold, and the deformable model is able to achieve more accurate results. Namely, the speed term F of equation (3.7) is defined here as

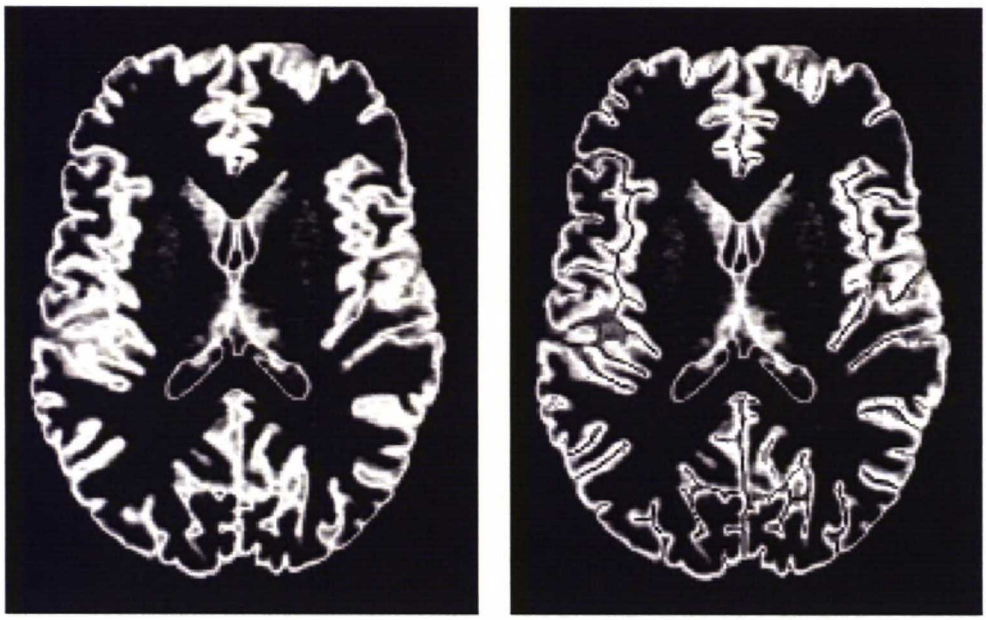
$$F(\mathbf{x}) = 1 - 0.9\mu_{\text{CSF}}(\mathbf{x}) \tag{3.9}$$

where $\mu_{\text{CSF}}(\mathbf{x})$is the CSF membership function.

After computing T of equation (3.7), the gradient ∇T is reevaluated using the centered finite difference operator, defined in the x-direction as

$$\frac{\partial \phi}{\partial x} \approx \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \tag{3.10}$$

and in the other directions in the same fashion. The shock points are then chosen as $\{\mathbf{x}|F(\mathbf{x})||\nabla T(\mathbf{x})|| \leq K\}$ based on the obtained approximation to the gradient. The search



(a) (b)

Figure 3.9 *(a) Original GM membership function, (b) GM membership function after ACE*

is limited to the set of points outside the inner cortical surface. K is a threshold less than 1, chosen empirically as 0.8.

Finally, the GM membership is modified at the found shock points. The new GM membership function is defined as

$$\mu'_{\mathrm{GM}}(\mathbf{x}) = \begin{cases} F(\mathbf{x})||\nabla T(\mathbf{x})||\mu_{\mathrm{GM}}(\mathbf{x}) & \text{if } \mathbf{x} \text{ is a shock point} \\ \mu_{\mathrm{GM}}(\mathbf{x}) & \text{otherwise} \end{cases} \tag{3.11}$$

3.6 Topology Preserving Level Set Method

3.6.1 Evolution Model

The general theory of geometric deformable models was briefly presented in section 2.4.4. In this section the used model is described in detail. The most important difference between the general level set method of Sethian et al. [48] and the implemented one is the handling of topology, which is explained later in section 3.6.4.

In a standard geometric deformable model, the evolving curve or surface $\Gamma(t)$ is embedded as the *zero level set* of the higher-dimensional level set function $\phi(\mathbf{x}, t)$

$$\Gamma(t) = \{\mathbf{x} | \phi(\mathbf{x}, t) = 0\} \tag{3.12}$$

The evolution is usually prescribed by a partial differential equation of the following form

$$\phi_t(\mathbf{x},t) = F_{\text{prop}}(\mathbf{x})||\nabla\phi(\mathbf{x},t)|| + F_{\text{curv}}(\mathbf{x})||\nabla\phi(\mathbf{x},t)|| + \vec{F}_{\text{adv}}(\mathbf{x}) \cdot \nabla\phi(\mathbf{x},t) \tag{3.13}$$

where F_{prop}, F_{curv} and $\vec{F}_{\text{adv}}$ are spatially varying speed terms. F_{prop} is an expansion or contraction speed in the normal direction, F_{curv} is the curvature term that depends on the intrinsic geometry of the surface and $\vec{F}_{\text{adv}}$ is the advection term, which represents an independent velocity field. (See Han et al. [23,24] and Sethian et al. [48])

By convention, the level set function $\phi(\mathbf{x},t)$ is initialized as a signed distance function to the initial surface $\Gamma(t=0)$

$$\phi(\mathbf{x},t=0) = \pm d(\mathbf{x}) \tag{3.14}$$

where $d(\mathbf{x})$ is the distance from $\mathbf{x}$ to $\Gamma(t=0)$, choosing negative values on the inside and positive on the outside of $\Gamma(t)$. If $\phi(\mathbf{x},t)$ is a signed distance function, it also has a property $||\nabla\phi(\mathbf{x},t)|| = 1$. After initialization, $\Gamma(t)$ is expressed as the zero level set of $\phi(\mathbf{x},t)$ as presented in equation (3.12) (see Han et al. [23,24] and Sethian et al. [47,48]). Computing the signed distance function can be done efficiently with the Fast Marching Method, which is discussed in section 3.5.2. The distances are computed to the initial surface that was generated in an earlier stage of the method.

As the level set function evolves, it drifts gradually away from the form of a signed distance function. To achieve accuracy and computational stability, the level set function is re-initialized periodically, again with the Fast Marching Method (see Osher et al. [38]). As the surface has evolved, the current zero level set of the implicit function $\phi(\mathbf{x},t)$ is used as the surface to which the distances are computed. The Fast Marching Method needs initial distance values near the evolving surface for initialization, and in CRUISE an isosurface was created at every recomputation of the signed distance function. The initial distances were computed from the generated isosurface. In the our implementation however, the isosurface method of CRUISE was not available, so a slightly different approach was chosen. The initialization was done by first selecting all voxels adjacent to the zero level set. This was done simply by finding grid points with positive values having neighbors with negative values, and vice versa. Grid points not selected this way were assigned a value of infinity or negative infinity, depending on whether the point was outside or inside the surface prior to this step. The selected points retained their values if their absolute value was less than 0.5, otherwise they were limited to 0.5 for positive and -0.5 for negative values.

In this implementation of the geometric deformable model, the two speed terms F_{prop} and F_{curv} of equation (3.13) are used in reconstructing all three cortical surfaces. F_{prop} is a signed pressure force computed from the fuzzy membership functions and F_{curv} is proportional to the mean curvature $\kappa(\mathbf{x})$ of the surface (see below equation 3.21). In the construction of the central surface, also the advection term $\vec{F}_{\text{adv}}$ is used to direct the surface towards the center of the cortex. The advection term is taken to be a GVF field $\vec{v}(\mathbf{x})$, introduced by Xu et al. [61] and used for cortical reconstruction in their later study [63]. The GVF is introduced in detail below in section 3.6.2.

These choices form the evolution equation to

$$\phi_t(\mathbf{x},t) = \omega_R R(\mathbf{x})||\nabla\phi(\mathbf{x},t)|| + \omega_\kappa \kappa(\mathbf{x},t)||\nabla\phi(\mathbf{x},t)|| + \omega_{\vec{v}}\vec{v}(\mathbf{x}) \cdot \nabla\phi(\mathbf{x},t) \tag{3.15}$$

where ω_R, ω_κ and $\omega_{\vec{v}}$ are weights that cause the terms to be emphasized differently. In this work, the first two weights are chosen as $\omega_R = 1$ and $\omega_\kappa = -0.02$ for all the surfaces. The weight of the advection term is chosen as $\omega_{\vec{v}} = 1$ in the reconstruction of the central surface, and for the other surfaces $\omega_{\vec{v}} = 0$, since the term is not applied in these cases.

Next we define the term $R(\mathbf{x})$ when constructing the different surfaces. For the sake of disambiguation, the term is referred to as $R_{\text{in}}(\mathbf{x})$, $R_{\text{cent}}(\mathbf{x})$ and $R_{\text{out}}(\mathbf{x})$ for the inner, central and outer cortical surfaces, respectively:

$$R_{\text{in}}(\mathbf{x}) = 2\mu'_{\text{WM}}(\mathbf{x}) - 1.40 \tag{3.16}$$

$$R_{\text{cent}}(\mathbf{x}) = \begin{cases} 0, & \text{if} ||2\mu'_{\text{WM}}(\mathbf{x})) + \mu'_{\text{GM}}(\mathbf{x}) - 1|| < 0.5 \\ 2\mu'_{\text{WM}}(\mathbf{x})) + \mu'_{\text{GM}}(\mathbf{x}) - 1 & \text{otherwise} \end{cases} \tag{3.17}$$

and

$$R_{\text{out}}(\mathbf{x}) = 2(\mu'_{\text{GM}}(\mathbf{x}) + \mu'_{\text{WM}}(\mathbf{x})) - 1.40, \tag{3.18}$$

where μ'_{WM} and μ'_{GM} are the modified WM and GM membership functions, respectively.

Following the definition of Osher et al. [38], the normal $\vec{N}$ of the surface $\phi(\mathbf{x}, t)$ at point $\mathbf{x}$ is defined as

$$\vec{N} = \frac{\nabla\phi(\mathbf{x}, t)}{||\nabla\phi(\mathbf{x}, t)||} \tag{3.19}$$

and the mean curvature $\kappa(\mathbf{x})$ of the interface is the divergence of the normal

$$\kappa(\mathbf{x}) = \nabla \cdot \vec{N} = \nabla \cdot \left(\frac{\nabla\phi(\mathbf{x}, t)}{||\nabla\phi(\mathbf{x}, t)||}\right) \tag{3.20}$$

and this can be expressed further as

$$\kappa(\mathbf{x}) = \frac{\begin{array}{c}\phi_{x_1}^2\phi_{x_2x_2} - 2\phi_{x_1}\phi_{x_2}\phi_{x_1x_2} + \phi_{x_2}^2\phi_{x_1x_1} + \phi_{x_1}^2\phi_{x_3x_3} - 2\phi_{x_1}\phi_{x_3}\phi_{x_1x_3} \\ + \phi_{x_3}^2\phi_{x_1x_1} + \phi_{x_2}^2\phi_{x_3x_3} - 2\phi_{x_2}\phi_{x_3}\phi_{x_2x_3} + \phi_{x_3}^2\phi_{x_2x_2}\end{array}}{||\nabla\phi||^3} \tag{3.21}$$

An additional constraint is applied in the deformable model to prevent overlapping of the three surfaces. This is made simply by restricting the level set function values to be lower than on a surface inside the evolving function. Formally, $\phi_{\text{out}}(\mathbf{x}, t) < \phi_{\text{cent}}(\mathbf{x}, t) < \phi_{\text{in}}(\mathbf{x}, t)$. This constraint guarantees the proper nesting of the three surfaces.

The numerical implementation of the evolution equation is discussed below in section 3.6.3.

3.6.2 Gradient Vector Flow

In the reconstruction of the central cortical surface, an external force field is needed to direct the deformable model to the center of the cortex. A gradient vector flow field is used for this purpose. GVF fields and their generalization are presented thoroughly in two papers by Xu et al. [61] and [62]. GVF fields are also applied to cortex reconstruction by Xu et al. in [63]. The most important details of GVF fields for this work will be discussed in this section.

GVF fields are dense vector fields derived from images by minimizing an energy functional. A deformable model using a GVF field is claimed to have two major advantages over traditional deformable models, namely insensitivity to initialization and the ability to move

into boundary concavities (see Xu et al. [61]). The GVF field is defined as the vector field $\vec{v}(x_1, x_2, x_3) = [u(x_1, x_2, x_3), v(x_1, x_2, x_3), w(x_1, x_2, x_3)]$ that is the equilibrium solution of the partial differential equation

$$\vec{v}_t(\mathbf{x}, t) = c\nabla^2\vec{v}(\mathbf{x}, t) - (\vec{v}(\mathbf{x}, t) - \nabla\mu'_{\mathrm{GM}}(\mathbf{x}))||\nabla\mu'_{\mathrm{GM}}(\mathbf{x})||^2 \tag{3.22}$$

where c is a weight set to 0.2, and $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2$ is the Laplacian operator, applied to each spatial component of $\vec{v}_t$ separately. The modified GM membership function $\mu'_{\mathrm{GM}}(\mathbf{x})$ acts as an edge map. When $||\nabla\mu'_{\mathrm{GM}}(\mathbf{x})||$ is large, the second term dominates the equation and $\vec{v}_t$ is assigned a value that is close to the gradient of the edge map. On the other hand, when $||\nabla\mu'_{\mathrm{GM}}(\mathbf{x})||$ is small, the first term affects the result more and causes the vector field to vary slowly in homogeneous regions.

The GVF field is found by solving the following Euler equations derived from (3.22)

$$c\nabla^2 u - (u - \frac{\partial}{\partial x}\mu'_{\mathrm{GM}}(\mathbf{x}))||\nabla\mu'_{\mathrm{GM}}(\mathbf{x})||^2 = 0 \tag{3.23}$$

$$c\nabla^2 v - (u - \frac{\partial}{\partial y}\mu'_{\mathrm{GM}}(\mathbf{x}))||\nabla\mu'_{\mathrm{GM}}(\mathbf{x})||^2 = 0 \tag{3.24}$$

$$c\nabla^2 w - (u - \frac{\partial}{\partial z}\mu'_{\mathrm{GM}}(\mathbf{x}))||\nabla\mu'_{\mathrm{GM}}(\mathbf{x})||^2 = 0 \tag{3.25}$$

Numerical implementation of the GVF field computation is discussed below in section 3.6.3. In figures 3.10(a)–(c) the computed GVF fields are shown in all three directions. As can be seen in figure 3.10(a), at the outer boundary of the cortex in x_1-direction the field has negative values on the left and positive values on the right side. With the chosen coordinate axis these values direct the evolving surface towards the center of the cortex, as expected. Figure 3.10(d) shows the magnitude of the vector field, and it is easy to notice that the vector field has the highest magnitude at cortex boundaries and low values at the middle of the cortex. In the reconstruction of the central cortical surface, the fuzzy clustering results are used to keep the surface within the cortex area and the GVF field directs the surface to the middle of the cortex.

3.6.3 Numerical Implementation

The implicit function ϕ is defined on a uniform Cartesian grid in the implementation of the geometric deformable model. Differential operators used in several equations need to be numerically discretized, which is done with finite difference techniques in this work. This section defines the different approximations used. First the basic operators are presented, followed by approximations used for different situations. The definitions are made only in the x_1-direction, but the approximations in other directions are obtained by symmetry. These definitions follow the ones presented in the book by Stanley Osher and Ronald Fedkiw [38].

First-order accurate forward difference $D^+\phi$ at grid point i is defined as

$$\frac{\partial\phi}{\partial x_1} = \frac{\phi_{i+1} - \phi_i}{\triangle x_1}, \tag{3.26}$$

where $\triangle x_1$ is the grid cell size in the x_1-direction. The first-order accurate backward difference $D^-\phi$ is defined as

$$\frac{\partial\phi}{\partial x_1} = \frac{\phi_i - \phi_{i-1}}{\triangle x_1} \tag{3.27}$$

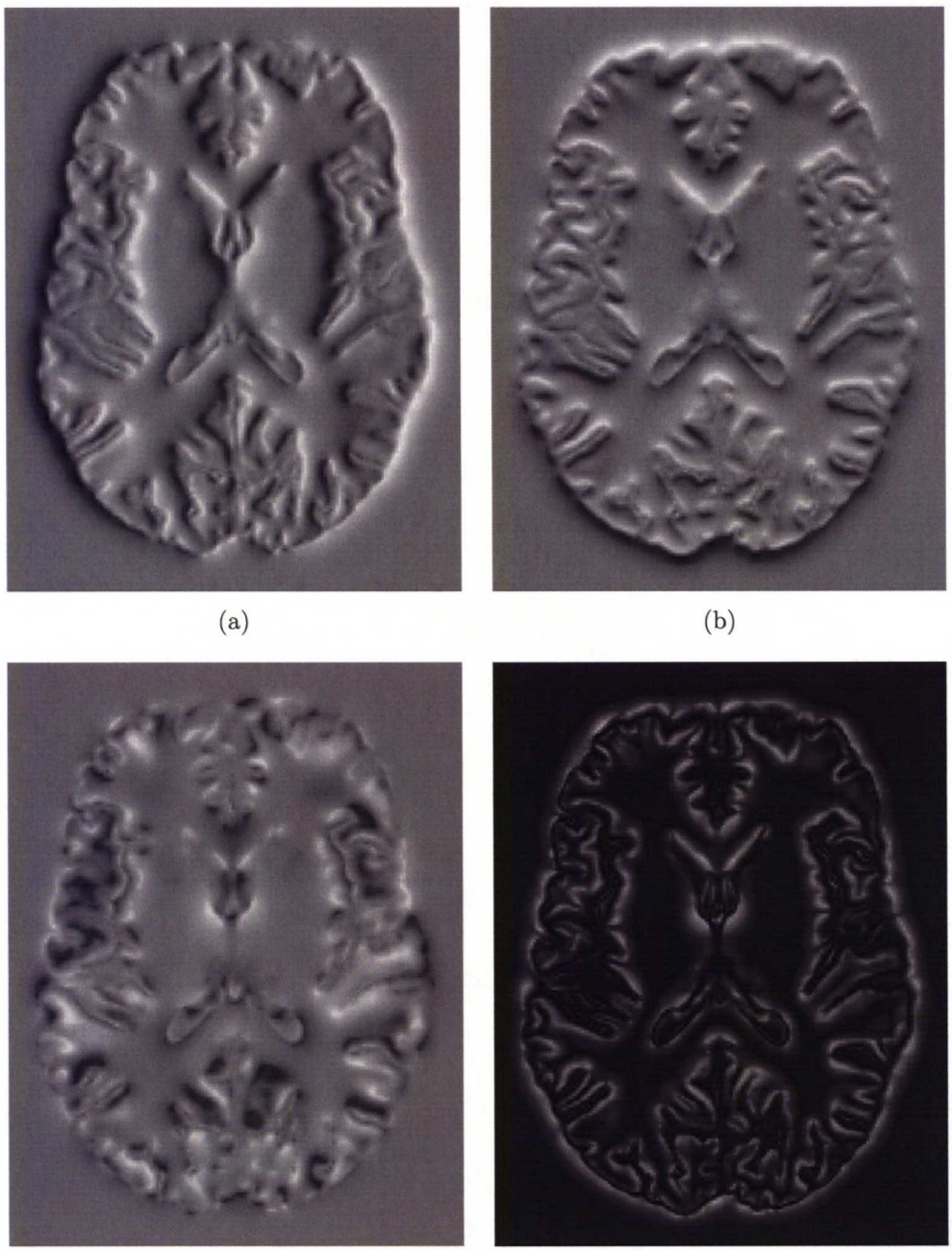


Figure 3.10 *GVF field in (a) x_1-direction, (b) x_2-direction, (c) x_3-direction and (d) sum of the absolute values of the three GVF fields. In (a), (b) and (c) dark areas correspond to negative values and light areas to positive values. The values with the same shade of gray as at image edges corresponds to zero. In (d) all values are non-negative, so that image intensity corresponds to directly to the strength of the vector field.*

and second-order accurate central difference $D^0\phi$ as

$$\frac{\partial \phi}{\partial x_1} = \frac{\phi_{i+1} - \phi_{i-1}}{2\triangle x_1}. \tag{3.28}$$

In a similar fashion, a second-order accurate finite difference formula for the second partial derivative of ϕ in the x_1-direction $D_x^+ D_x^- \phi$ (or equivalently $D_x^- D_x^+ \phi$) is defined as

$$\frac{\partial^2 \phi}{\partial x_1^2} = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\triangle x_1^2}. \tag{3.29}$$

A second order accurate finite difference formula for $\phi_{x_1 x_2}$ is given by $D_{x_1}^0 D_{x_2}^0 \phi$ (equivalently $D_{x_2}^0 D_{x_1}^0 \phi$), which means taking the central difference (3.28) first in x_2-direction (x_1-direction) and then taking the central difference of the result in the x_1-direction (x_2-direction).

In the level set equation (3.13) all three terms need different approximations to the gradient. When handling moving fronts, one needs to consider the movement of the front to be able to choose an accurate approximation. One cannot simply use only one of the first-order difference approximations (3.26, 3.27, or 3.28), since that approach would fail according to Osher et al. [38]. A more accurate approximation is obtained with methods such as upwind differencing, which takes the direction in which the front is advancing in to account.

First we look at the case of monotonically advancing fronts, which is represented here with a simple convection (or advection) equation

$$\phi_t + \vec{V} \cdot \nabla \phi = 0 \tag{3.30}$$

for the implicit function ϕ, where $\vec{V} = [u(x_1, x_2, x_3), v(x_1, x_2, x_3), w(x_1, x_2, x_3)]$ is a velocity field defined at every grid point $[x_1, x_2, x_3]$. This corresponds to the advection term in (3.13). At a point in time t^n, let $\phi^n = \phi(t^n)$ represent the current values of ϕ. When ϕ is updated in time, a new value is found for every grid point after some time increment $\triangle t$. The new values are denoted as $\phi^{n+1} = \phi(t^{n+1})$, where $t^{n+1} = t^n + \triangle t$. Equation (3.30) can then be written as

$$\frac{\phi^{n+1} - \phi^n}{\triangle t} + u^n \phi_x^n + v^n \phi_y^n + w^n \phi_z^n = 0. \tag{3.31}$$

Considering a one-dimensional example and focusing on a single grid point, equation (3.31) becomes

$$\frac{\phi_i^{n+1} - \phi_i^n}{\triangle t} + u_i^n (\phi_x)_i^n = 0, \tag{3.32}$$

where $(\phi_x)_i$ denotes the spatial derivative of ϕ at the point x_i. Now, if $u_i > 0$, the values of ϕ are moving from left to right and if $u_i < 0$, the values are moving from right to left. For an accurate approximation, one should use the backward difference (3.27) when $u_i > 0$ and forward difference (3.26) when $u_i < 0$. This method is called upwind differencing.

In the case where the front is expanding or contracting in the normal direction, simple upwind differencing is not adequate when $D^+\phi$ and $D^-\phi$ differ in sign. This happens in the vicinity of a sonic point, where $\phi_x = 0$, or at the location of a shock wave where opposing fronts meet. Using the Godunov scheme enables the handling these situations.

The level set equation for a front advancing in the normal direction is

$$\phi_t + a||\nabla \phi|| = 0, \tag{3.33}$$

where a is a constant. When $a > 0$ the interface moves in the normal direction, and when $a < 0$ the interface moves opposite to the normal direction. The choice between ϕ_x^+ and ϕ_x^- depends on the signs of the products $a\phi_x^+$ and $a\phi_x^-$. There are four different possible cases:

1. $a\phi_x^+ > 0$ and $a\phi_x^- > 0$,
2. $a\phi_x^+ < 0$ and $a\phi_x^- < 0$
3. $a\phi_x^+ \geq 0$ and $a\phi_x^- \leq 0$
4. $a\phi_x^+ \leq 0$ and $a\phi_x^- \geq 0$

Now we look at how the Godunov scheme instructs to deal with these cases. The first two are the simplest ones since basic upwind differencing may be used, ϕ_x^- for case 1 and ϕ_x^+ for case 2. Case 3 corresponds to an expanding flat region, so $\phi_x = 0$ is chosen. In the last case of the shock wave caused by fronts meeting, either ϕ_x^+ or ϕ_x^- is chosen, depending on which gives the largest magnitude for $a\phi_x^+$. This is also a reasonable choice, since the front corresponding to the difference with the larger magnitude is supposed to arrive first at the observation point.

Finally there is the curvature term which is different from the advection and expansion terms presented above, since the information is not coming from a certain direction, it is rather dependent on the surrounding neighborhood. For this reason, the gradient is approximated using the central difference approximation (3.28).

For the gradient vector field, we need to solve the three equations (3.23),(3.24) and (3.25). Since also in this case all three dimensions are treated in a similar fashion, we present here the numerical implementation for only one direction. The equation (3.23) is treated now as a function of time and the solution is obtained by solving

$$\begin{aligned} u_t(x_1,x_2,x_3,t) = \mu\nabla^2 u(x_1,x_2,x_3,t) - [u(x_1,x_2,x_3,t) - f_{x_1}(x_1,x_2,x_3)] \\ \cdot[f_{x_1}(x_1,x_2,x_3)^2 + f_{x_2}(x_1,x_2,x_3)^2 + f_{x_3}(x_1,x_2,x_3)^2], \end{aligned} \tag{3.34}$$

which can be rewritten as

$$u_t(x_1,x_2,x_3,t) = \mu\nabla^2 u(x_1,x_2,x_3,t) - b(x_1,x_2,x_3)u(x_1,x_2,x_3,t) + c^1(x_1,x_2,x_3) \tag{3.35}$$

where

$$b(x_1,x_2,x_3) = f_{x_1}(x_1,x_2,x_3)^2 + f_{x_2}(x_1,x_2,x_3)^2 + f_{x_3}(x_1,x_2,x_3)^2 \tag{3.36}$$

and

$$c^1(x_1,x_2,x_3) = b(x_1,x_2,x_3)f_{x_1}(x_1,x_2,x_3). \tag{3.37}$$

Since $b(x_1,x_2,x_3)$ and $c^1(x_1,x_2,x_3)$ are not dependent on the time variable, they can be computed at the start and the same values can then be used for the rest of the computation. The central difference operator (3.28) is used in this implementation. The Laplacian operator is then approximated as

$$\begin{aligned} \nabla^2 u = \tfrac{1}{\triangle x_1 \triangle x_2 \triangle x_3}(u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j,k+1} + u_{i-1,j,k} + u_{i,j-1,k} \\ + u_{i,j,k-1} - 6u_{i,j,k}) \end{aligned} \tag{3.38}$$

and

$$u_t = \frac{1}{\triangle t}(u_{i,j,k}^{n+1} - u_{i,j,k}^{n}). \tag{3.39}$$

Using the above approximations, the solution to GVF is found as

$$\begin{aligned} u_{i,j,k}^{n+1} = (1 - b_{i,j,k}\triangle t)u_{i,j,k}^{n} + r(u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j,k+1} + u_{i-1,j,k} \\ +u_{i,j-1,k} + u_{i,j,k-1} - 6u_{i,j,k}) + c_{i,j,k}^{1}\triangle t \end{aligned} \tag{3.40}$$

where

$$r = \frac{\mu\triangle t}{\triangle x_1 \triangle x_2 \triangle x_3}. \tag{3.41}$$

3.6.4 Topology Preservation

As discussed before, it is crucial for a cortex reconstruction method to result with the anatomically correct spherical topology. This requirement poses a challenge for a geometric deformable model, which is not designed to control the topology of the result. In this work, the problem is addressed by initializing the level set function with a topologically correct surface and preserving the topology by preventing topological changes.

Grid points with negative or zero values of the level set function correspond to the inside of the object and positive values correspond to the outside. This means that topological changes are only possible when the value at a grid point changes sign. The idea for preserving topological properties of the object is very simple; if a grid point value is about to change sign, we check if the change will alter the topology of the object. Then the sign change is allowed only if the topology is left unchanged. As noted above, a point is called topologically *simple* if the topology of the object is unaltered when the point is added or removed. In this section we discuss how to distinguish these simple points from other points, and how this is incorporated in the geometric deformable model. Combining the geometric deformable model with topological preservation in the presented way was originally introduced by Han et al. [23].

The following definitions of digital topology are from Bertrand et al. [8] and Han et al. [23]. On a digital grid, a point $\mathbf{x} \in \mathbb{Z}^3$ is defined by $\mathbf{x} = (x_1, x_2, x_3)$, $x_i \in Z$. Consider the two distance functions

$$D_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{3} |x_i - y_i| \tag{3.42}$$

and

$$D_\infty(\mathbf{x}, \mathbf{y}) = \max_{i=1,2,3} |x_i - y_i| \tag{3.43}$$

The associated neighborhoods of $\mathbf{x}$ are

$$V_1^i(\mathbf{x}) = \{\mathbf{y} | D_1(\mathbf{x}, \mathbf{y}) \leq i\} \tag{3.44}$$

and

$$V_\infty^i(\mathbf{x}) = \{\mathbf{y} | D_\infty(\mathbf{x}, \mathbf{y}) \leq i\} \tag{3.45}$$

The commonly used neighborhoods are the 6-neighborhood $N_6(\mathbf{x}) = V_1^1(\mathbf{x})$, the 26-neighborhood $N_{26}(\mathbf{x}) = V_\infty^1(\mathbf{x})$ and the 18-neighborhood $N_{18}(\mathbf{x}) = V_1^2(\mathbf{x}) \cap V_\infty^1(\mathbf{x})$. We denote a general

n-neighborhood of $\mathbf{x}$ by $N_n(\mathbf{x})$ and the set comprising of the neighborhood of $\mathbf{x}$ with $\mathbf{x}$ removed by $N_n^*(\mathbf{x})$. Two points $\mathbf{x}$ and $\mathbf{y}$ are said to be n-adjacent if $\mathbf{y} \in N_n^*(\mathbf{x})$. An n-path is a sequence of points $\mathbf{x}_0, .., \mathbf{x}_k$, with $\mathbf{x}_i$ n-adjacent to $\mathbf{x}_{i-1}$ for $i = 1, .., k$.

An object $X \subset \mathbb{Z}^3$ is said to be n-connected if for any two points of X, there is an n-path in X between these two points. The set of all the n-connected components of X is noted $C_n(X)$. The set of all the n-connected components of X n-adjacent to a point $\mathbf{x}$ is noted $C_n^a(\mathbf{x}, X)$.

The topology of a digital object depends on a pair of digital connectivities, one for the object X and one for the background $\bar{X}$. To avoid a connectivity paradox, different connectivities n and $\bar{n}$ have to be selected for X and $\bar{X}$. The 6-connectivity for X is associated with the 18- or 26-connectivity for $\bar{X}$, and vice versa. It is necessary to distinguish between the 6-connectivity associated with the 18-connectivity and the 6-connectivity associated with the 26-connectivity. For this reason, we note 6^+-connectivity as associated with 18-connectivity. So the different possible pairs for digital connectivity are $(n, \bar{n}) = (6, 26), (26, 6), (6^+, 18)$ and $(18, 6^+)$.

Next we define $N_n^k(\mathbf{x}, X)$, the *geodesic n-neighborhood* of $\mathbf{x}$ inside X of order k recursively by

$$N_n^1(\mathbf{x}, X) = N_n^*(\mathbf{x}) \cap X \tag{3.46}$$

and

$$N_n^k(\mathbf{x}, X) = \cup\{N_n(\mathbf{y}) \cap N_{26}^*(\mathbf{x}) \cap X, \mathbf{y} \in N_n^{k-1}(\mathbf{x}, X)\} \tag{3.47}$$

Some geodesic neighborhoods have the following relation to more commonly used neighborhoods:

$$C_6[N_6^2(\mathbf{x}, X)] = C_6^a[\mathbf{x}, N_{18}^*(\mathbf{x}) \cap X] \tag{3.48}$$

$$C_{18}[N_{18}^2(\mathbf{x}, X)] = C_{18}^a[\mathbf{x}, N_{26}^*(\mathbf{x}) \cap X] \tag{3.49}$$

$$C_{26}[N_{26}^1(\mathbf{x}, X)] = C_{26}^a[\mathbf{x}, N_{26}^*(\mathbf{x}) \cap X] \tag{3.50}$$

Finally we define the *topological numbers* relative to the point $\mathbf{x}$ and the set X, where $\#$ denotes the cardinality of a set

$$T_6(\mathbf{x}, X) = \#C_6[N_6^2(\mathbf{x}, X)] \tag{3.51}$$

$$T_{6^+}(\mathbf{x}, X) = \#C_6[N_6^3(\mathbf{x}, X)] \tag{3.52}$$

$$T_{18}(\mathbf{x}, X) = \#C_{18}[N_{18}^2(\mathbf{x}, X)] \tag{3.53}$$

$$T_{26}(\mathbf{x}, X) = \#C_{26}[N_{26}^1(\mathbf{x}, X)] \tag{3.54}$$

A point $\mathbf{x}$ is an n-simple point if and only if $T_n(x, X) = 1$ and $T_{\bar{n}}(x, \bar{X}) = 1$, where n and $\bar{n}$ are a suitable pair of connectivities as discussed above. If the level set function value of a point is about to change sign, these two topological numbers are computed. If the point is simple, the level set function value is changed normally. For a non-simple point, a value ϵ is assigned if the previous value was positive, or $-\epsilon$ if the previous value was negative, where ϵ is a small positive number.

One downside of the method is that the result changes depending on the order in which the grid points are visited. The values of the level set function are all computed only from values of the previous time step, but when deciding if a point is simple or not, also the grid point values computed on the current time step have to be taken in to account. Otherwise, several points might change sign at the same time and alter the topology even if they both were simple points in the last time step. This issue can be avoided to some extent by advancing the front slowly. In this way less neighboring points are changing sign at the same time step. The problem was also addressed by Han et al. [23], and they concluded that the resulting difference is trivial when changing the order in which points are updated.

3.6.5 Narrow Band Implementation

Constructing the signed distance function and evolving the level set function within the whole image volume is computationally expensive. Moreover, the level set function does not stay as a signed distance function as it evolves, and it has to be reinitialized periodically to achieve accurate results. Because of this, computing values for points far away from the zero level set is not useful, and this is an obvious possibility to reduce the computational expense of the method.

The narrow band approach was introduced by Adalsteinsson and Sethian in [1]. In their method, the signed distance function is constructed only on a narrow band around the evolving contour, and the rest of the grid points are ignored. If the evolving contour (the zero level set) comes close to the edge of the narrow band, the narrow band is reconstructed around the current zero level set. The location of the narrow band does not change while the level set function is evolving.

The original approach of Adalsteinsson and Sethian tries to minimize the need to reconstruct the narrow band, and does it only when the zero level set reaches the band edge. As the contour comes very close to the edge of the narrow band, special care has to be taken as derivatives are not well-defined at the edge.

A simplified implementation for reinitialization was chosen in this work. The narrow band is reconstructed after a certain amount of evolution iterations, regardless of the zero level set location in relation to the band edge. The narrow band is constructed sufficiently wide, so that the zero level set never reaches the edge before the band is reconstructed. Also the level set function is always reinitialized as a signed distance function every time the narrow band is reconstructed. This way the edge points do not have a big influence on the evolution of the zero level set and the approximation to the gradient at the edge points is not so important.

Points located at the edge of the narrow band are assigned static values for the evolution equation (3.15) variables that depend on neighboring values. Namely this means $||\nabla\phi(\mathbf{x},t)|| = 1$ and $\kappa(\mathbf{x}) = 0$. In cases where the advection term is applied, it is also set to zero. In other words, the evolution of points at the narrow band edge does not depend on neighboring points. It is still useful to evolve these edge points instead of assigning static level set function values to them, since they affect the evolution of their neighbors within the narrow band. The chosen static value for the gradient magnitude is a reasonable choice,

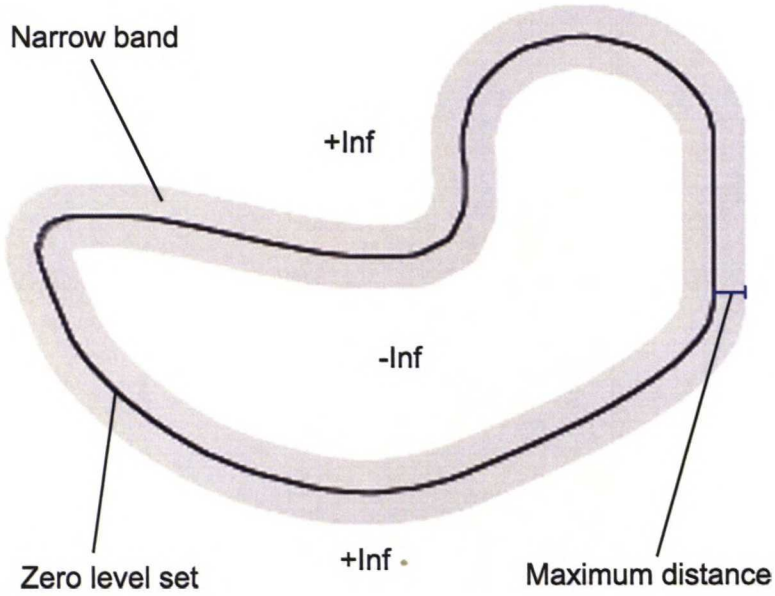


Figure 3.11 *The implicit level set function $\phi(\mathbf{x}, t)$ is defined only on the narrow band. The evolving surface is represented as the zero level set of the level set function and is located at the center of the narrow band after every reinitialization. Values outside the narrow band are positive or negative infinity, depending on whether they are inside or outside the surface.*

since in a perfect signed distance function it would have this value everywhere. Keeping unity as the value is the same as assuming the level set function to stay as a signed distance function. The mean curvature value can be either positive or negative depending on the neighborhood, so it is simply assigned a zero value, which removes the influence of the term in the evolution.

The chosen implementation of static values for the evolution equation variables at the narrow band edge points would most likely cause inaccurate results or at least additional iterations if the zero level set was allowed to reach the band edge. However, in this work the narrow band is reconstructed and the level set function reinitialized at regular intervals, well before the zero level set reaches the edge. This way, the chosen solution is assumed to give the same results as the original method of Adalsteinsson and Sethian. However, in the chosen implementation the narrow band is reconstructed more times than necessary, and reducing this frequency would reduce computational expense.

In this work, the narrow band was constructed using the standard Fast Marching Method (see section 3.5.2) by simply stopping the computation when the minimum candidate distance of the Fast Marching Method reaches a certain value. Distinguishing between points inside and outside the narrow band is done by classifying points with a distance computed this way inside the band, and rest of the grid points outside.

3.6.6 Surface Generation

In the current implementation, the final isosurfaces were generated with the MATLAB isosurface tool using the zero level set of the converged level set function $\phi(\mathbf{x}, t)$. However, this isosurface method does not guarantee preservation of topology. For this reason a topology preserving isosurface method is required, such as the one by Han et al. [23]. Work on an implementation is already underway in our research group.

Chapter 4

Results

4.1 Data and Evaluation Methods

The reconstruction method was tested with image volumes from the IXI database provided by Imperial College of Science Technology & Medicine and University College London, funded by UK EPSRC under grants GR/S21526/01 and GR/S21533/01 and accessible at http://www.ixi.org.uk/. The used images are standard anatomical T1-weighted volumes (MP-RAGE) from normal subjects between the age of 20 and 80 years. The images were obtained in sagittal slices with in-slice pixel resolution of 0.9375 mm and 1.1999 mm between slices.

For this study, 85 image volumes with the smallest database ID values (2–102) were downloaded and the cortex reconstruction method was applied to them (three of these images were discarded, of which two volumes were corrupted, IDs 12 and 50, and in one case the data quality was radically worse than average, causing the preprocessing phase to fail, ID 35).

The results were initially inspected visually. After this, for 30 of the images with smallest ID values (2–45), a landmark study was conducted to get numerical error estimates for the reconstruction method. In the following sections, the results are presented together with a brief analysis of the performance of different stages.

The software written in this work and used to generate the surfaces from the fuzzy clustering results takes between 30 and 40 minutes per image to execute. Time estimates for earlier stages are presented below. The total time taken from raw image to reconstructed surfaces is roughly between 50 and 65 minutes. The reconstructions were computed on a desktop computer with an Intel Core 2 Quad Q6600 2.4 GHz processor and 4 Gb of memory. The software uses currently only a single core of the processor.

4.2 Visual Inspection

The reconstruction method worked expectedly on all cases, although some inaccuracies were visible in a few reconstructions. In this section, the performance of each step of the method is briefly analyzed.

The preprocessing phase caused most difficulties in the reconstruction process. More specifically, skull-stripping required manual interaction in several cases, particularly in volumes where intensity inhomogeneities were strong. This was caused by thresholding using a single constant value, which understandably failed in regions where the average image intensity differed from the average. The cerebrum extraction from the skull-stripped volume worked well every time. In some cases small parts of the cerebellum remained in the volume, but this was considered acceptable for the reconstruction method. The preprocessing took between 3 to 15 minutes per image, depending on the performance of the skull-stripping stage.

The fuzzy clustering stage confronted problems in some volumes with low contrast (see figure 4.1). In these cases, the clustering did not converge and the value of the neighborhood smoothing term (β in equation 3.2) was decreased to achieve convergence. In practice, every volume was segmented first using the value $\beta = 0.05$, iterating until the FANTASM cost function J_{FANTASM} reached a value of 0.01 or less. If the cost function did not reach this value after 50 iterations, the smoothing term β was assigned a new value of 0.01 and the clustering was performed again. Out of total 82 cases, in 22 the clustering did not converge with the higher value for β. From the 22 cases clustered with the lower value, convergence was still not achieved in 3 cases, where the reconstruction was continued using clustering results after 50 iterations. The fuzzy clustering took from 2 to 4 minutes, depending on the convergence.

In 50 cases, the WM membership function was manually edited before ventricle filling as described in section 3.3. This took from 1 to 3 minutes per image. The ventricle filling performed adequately almost every time, but manual interaction was also required afterward in most cases to remove parts where the region growing had advanced to unwanted areas. This was caused by holes in the binary WM membership function. These holes occur in thin sections of the WM membership function where misclassification of a voxel has caused it to have a low membership value and therefore it is part of the background in the binarized volume. The manual editing was performed on 55 volumes, and it took between 1 and 5 minutes, depending mostly on the number and size of the holes responsible for unwanted results in the binarized WM membership function. The used ventricle filling method is not robust enough for large-scale use. However, in its current interactive form it still provides savings in time compared to performing the stage manually.

The GM membership editing stage worked well in general but some unwanted behavior occurred at locations where cortical folds were not very tight, as shown in figure 4.2(c) and (d). In these regions, the advancing front of the fast marching method had several shock points, which caused the GM membership to be edited on a relatively large area, often partially in an unwanted location. This induced some inaccuracy in the final results. Also some displacement of the GM membership editing was noted in locations where the original image data showed only weak signs of CSF between banks of a cortical fold, shown

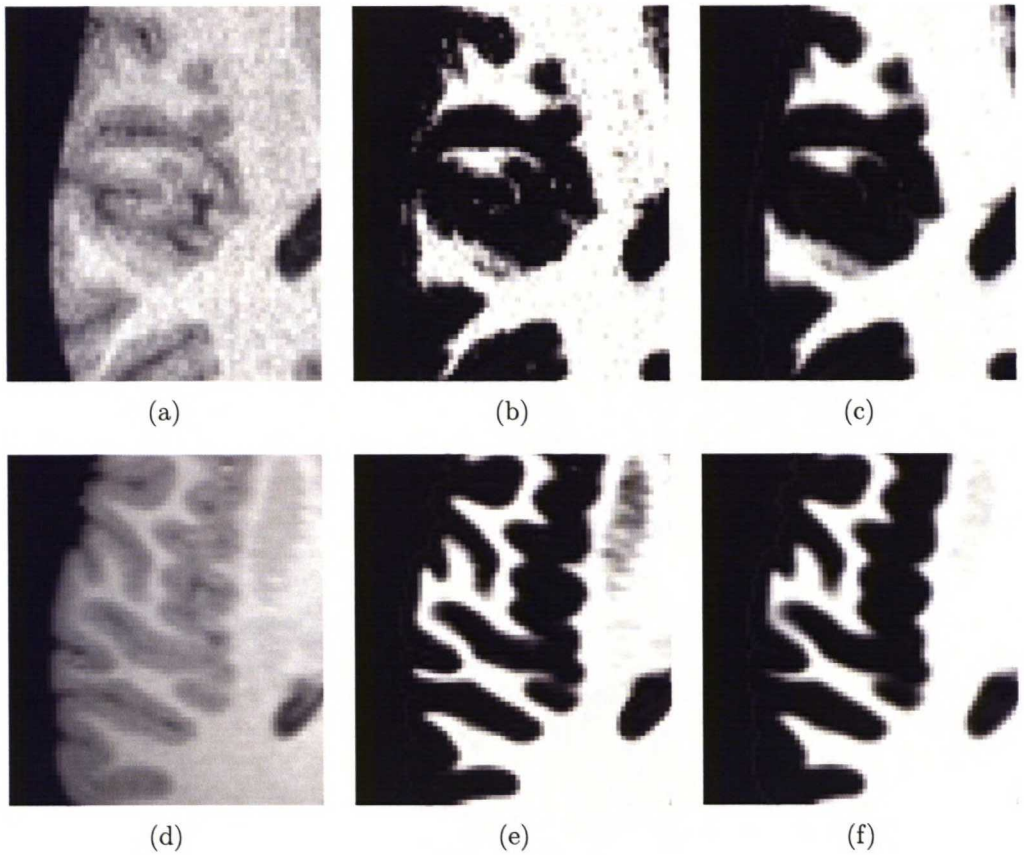


Figure 4.1 *Detailed view of fuzzy clustering results with different values for smoothing term β for two subjects. Subject 1: (a) original image, (b) β=0.01, (c) β=0.05. Subject 2: (d) original image, (e) β=0.01, (f) β=0.05.*

in figure 4.2(e) and (f). This is assumed to be caused by the fuzzy clustering neighborhood normalization, which hides noise but sometimes also subtle information of CSF within the cortical folds.

All the subsequent stages worked expectedly in every case. In figures 4.3 and 4.4 typical results of the segmentation are shown, and in figure 4.5 the reconstructions of the three cortical surfaces.

4.3 Landmark Study

The landmark study was conducted using 30 image volumes. Since the landmark study was conducted with skull-stripped volumes rather than original raw images in the works of Han et al. [24] and Tosun et al. [56], skull-stripped volumes were also used in this study to make the results comparable. From each volume, 288 landmark points were selected, bringing the total number to 8640. The landmarks were chosen so that 96 points were selected for each of the three surfaces, of which 32 correspond to geometry of sulcal fundi, 32 to sulcal banks

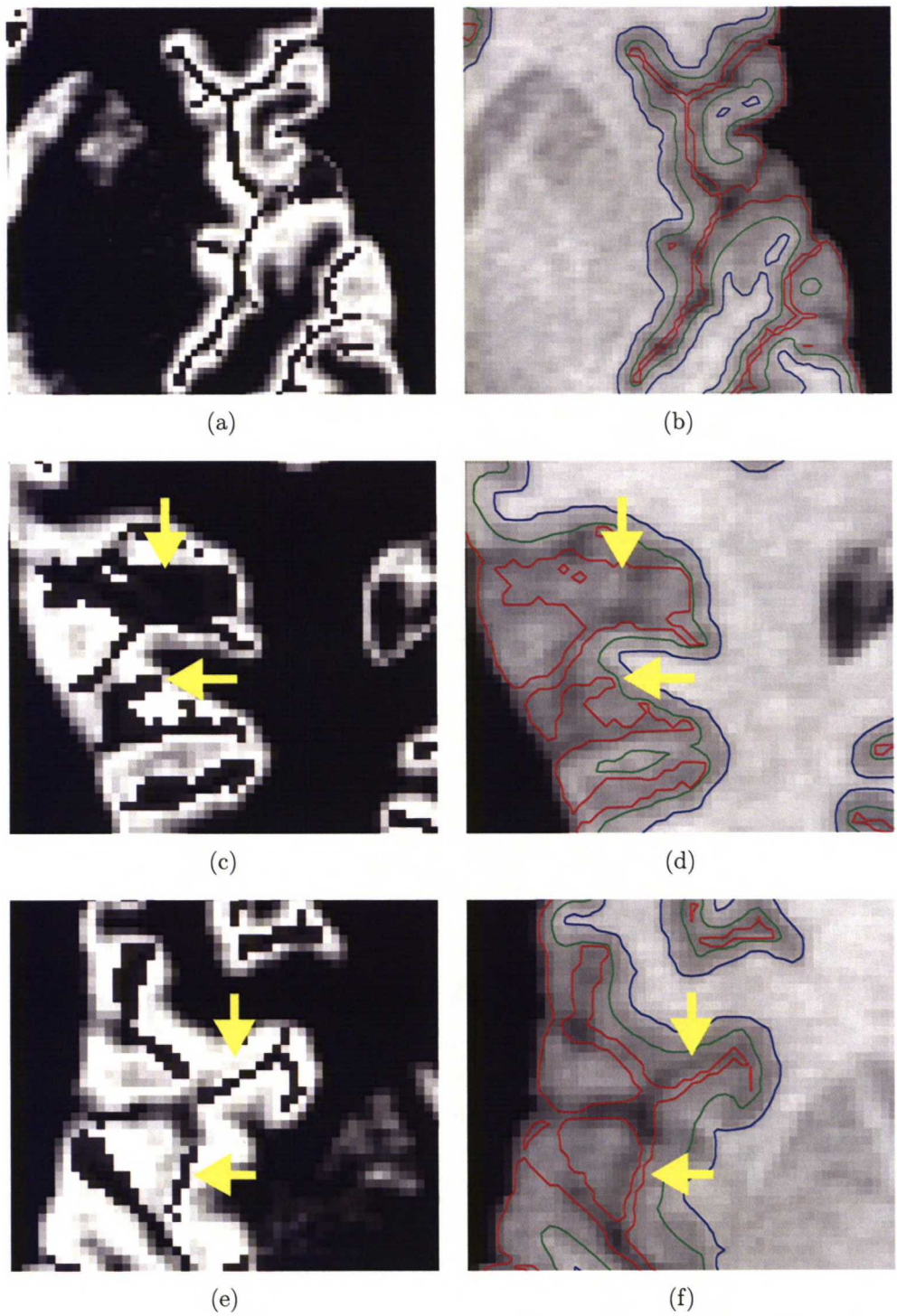


Figure 4.2 *Detailed views of ACE (GM enhancement) performance. In tight sulci where the distance between opposing sulcal banks is small, the algorithm works well: (a) ACE result, (b) reconstructed surfaces on the original image. Examples of problems associated with ACE are large edited plains and advancing in undesired areas (c),(d) and also displacement (e),(f).*

and the remaining 32 to gyral crowns. Sulcal fundus lies at the bottom of the cortical fold, sulcal bank at the side of the fold and gyral crown at the top of the fold (see Han et al. [24]).

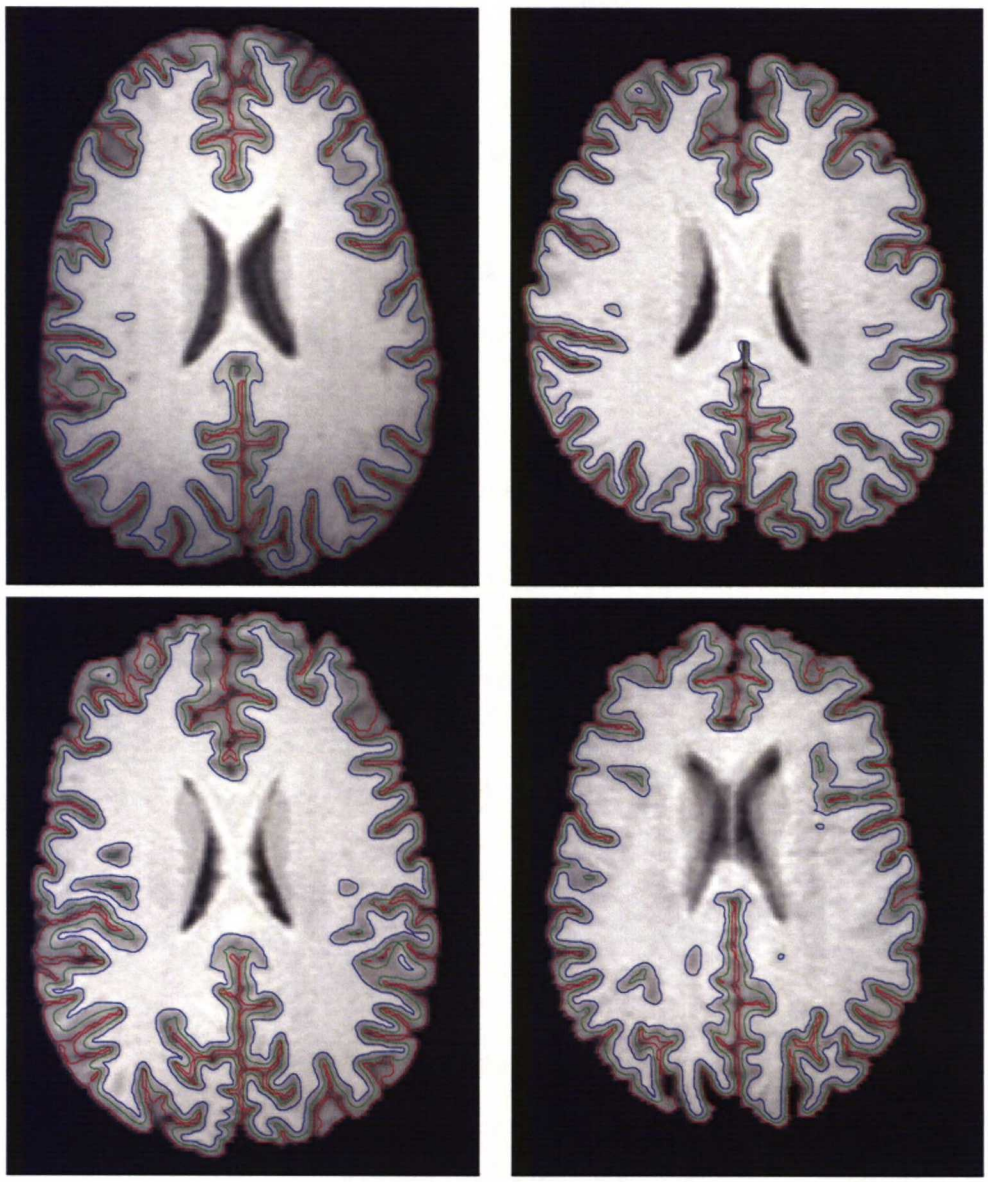


Figure 4.3 *Cross-sectional view of four subjects, with contours representing the three reconstructed surfaces (blue: inner, green: central, red: outer).*

The landmark points were selected by a single rater.

For the landmark study a program was written in MATLAB for selecting points in the image volume. The images were interpolated to isotropic voxels and viewed as axial cross-sections. The landmark points were selected for one cortical surface at a time. The image volume was divided equally in four sections (see figure 4.6), of which one was shown at a time and six landmark points were selected, two points for each cortical geometry type, as described above. For each image, the program started approximately from the putamen region level, and showed a view of the first section. After the six landmark points were selected, the next view was shown from the second section. With each new view the slice level was increased by two, so that each set of points from the same section were eight slices apart. This way a large part of the image volume was included in the landmark study. The same process was

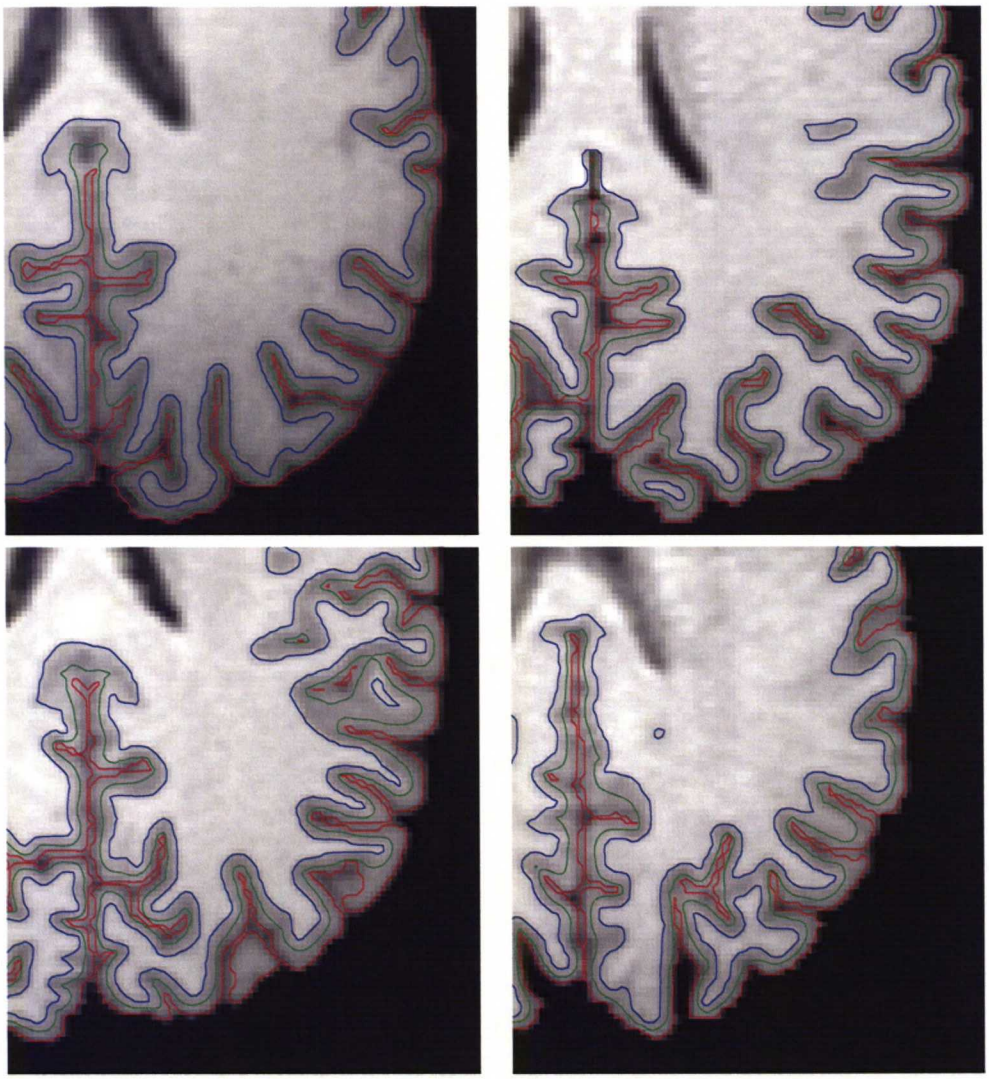


Figure 4.4 *Zoomed view of four subjects, with contours representing the three reconstructed surfaces (blue: inner, green: central, red: outer).*

done for all three surfaces, but the starting axial slice was changed by five levels, so that the points were selected from different parts of the volume for the three surfaces. Examples of chosen landmark points are shown in figure 4.7.

For each landmark point the minimum distance to the corresponding isosurface was computed with a program by MSc. Mikko Lilja. The computed positive distance values were changed to negative if the point was selected from inside the reconstructed cortical surface. A summary of the landmark study results is presented in table 4.1, and the full results in appendix A. In the tables the mean values of signed and absolute distances in millimeters are presented together with their standard deviations. Also the percentage of landmark points outside absolute distance value of 1 mm and 2 mm are shown.

The average absolute distance error measure for the inner cortical surface was 0.63 mm, with a standard deviation of 0.53 mm. The error measures at sulcal banks were slightly lower than for other geometries. There was a significant outward bias in the reconstruction, with

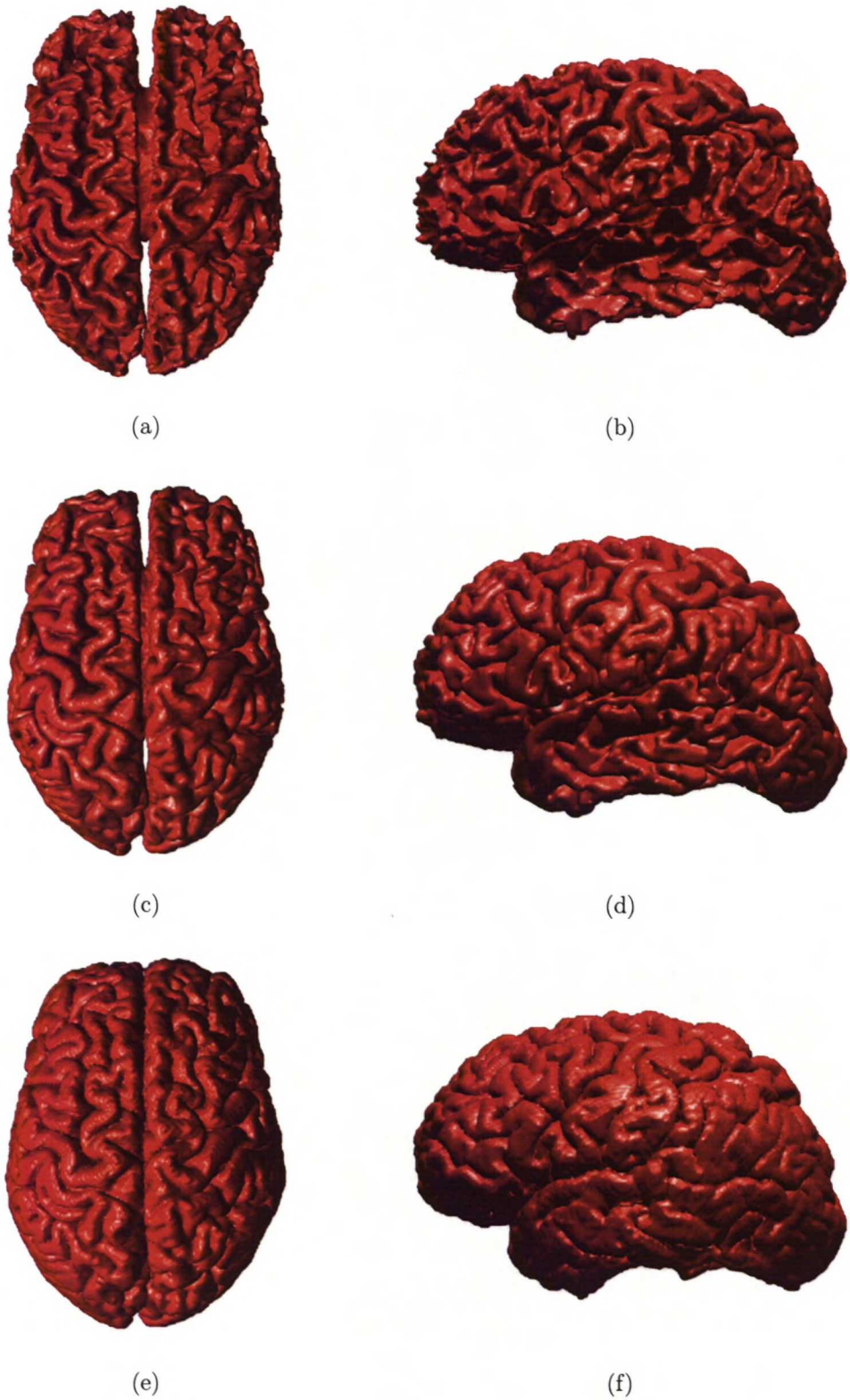


Figure 4.5 *3D-representations of the three reconstructed cortical surfaces, (a),(b) inner surface, (c),(d) central surface, (e),(f) outer surface*

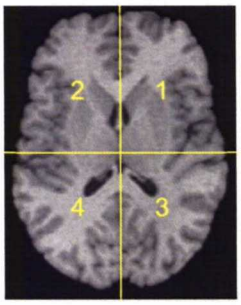


Figure 4.6 *Four sections used in the landmark point selection.*

the average signed distance at −0.57 mm. Table A.1 in Appendix A shows that the inner surface average signed distance was negative for every single volume in the study. The inner surface reconstruction was the most robust of the surfaces, with 1.46% of absolute errors larger than 2 mm.

Compared to measures of the inner surface, the central surface had similar average absolute error (0.67 mm), but the proportion of large errors with absolute distance larger than 2 mm was 2.85%, which is notably about double of the same measure for the inner surface. Average signed distance measure of −0.03 mm shows that there was no apparent bias for the central surface.

The reconstruction method performed best on average for the outer cortical surface, with an average absolute error of 0.42 mm. Highest error values were obtained at sulcal fundi, but the absolute error measure (0.53 mm) is still lower than the average error measures at any cortical geometry for either of the two other surfaces. Sulcal fundi had clearly a higher occurrence of large errors (3.02%) than at other cortical geometries. There was no notable bias for the outer surface, with a signed distance of −0.16 mm on average. The results are discussed in more detail in chapter 5.

4.4 Comparison of Landmark Study Results

The landmark study results were compared to results in the original work of Han et al. [24] and the more extensive study conducted by Tosun et al. [56]. Results of the three studies are shown in table 4.2. The work of Tosun and our implementation had similar parameter values for the geometric deformable model, with target fuzzy clustering membership values (see equation 3.16) of 0.70 in our study and 0.69 for Tosun. In the work of Han slightly different values were used, with 0.50 as the target value.

For the inner and outer cortical surfaces, the previous two studies had used only two image volumes, whereas 30 were used in our work. In the previous studies 10 landmarks were picked close to each other from each predefined view, choosing a total of 420 or 330 for each surface, whereas in this study single points were selected from random views with a total of 96 landmarks for each surface. Like this study, Han et al. [24] used a single rater, whereas Tosun et al. [56] used 9 raters for the test data. Tosun et al. [56] did not include a study

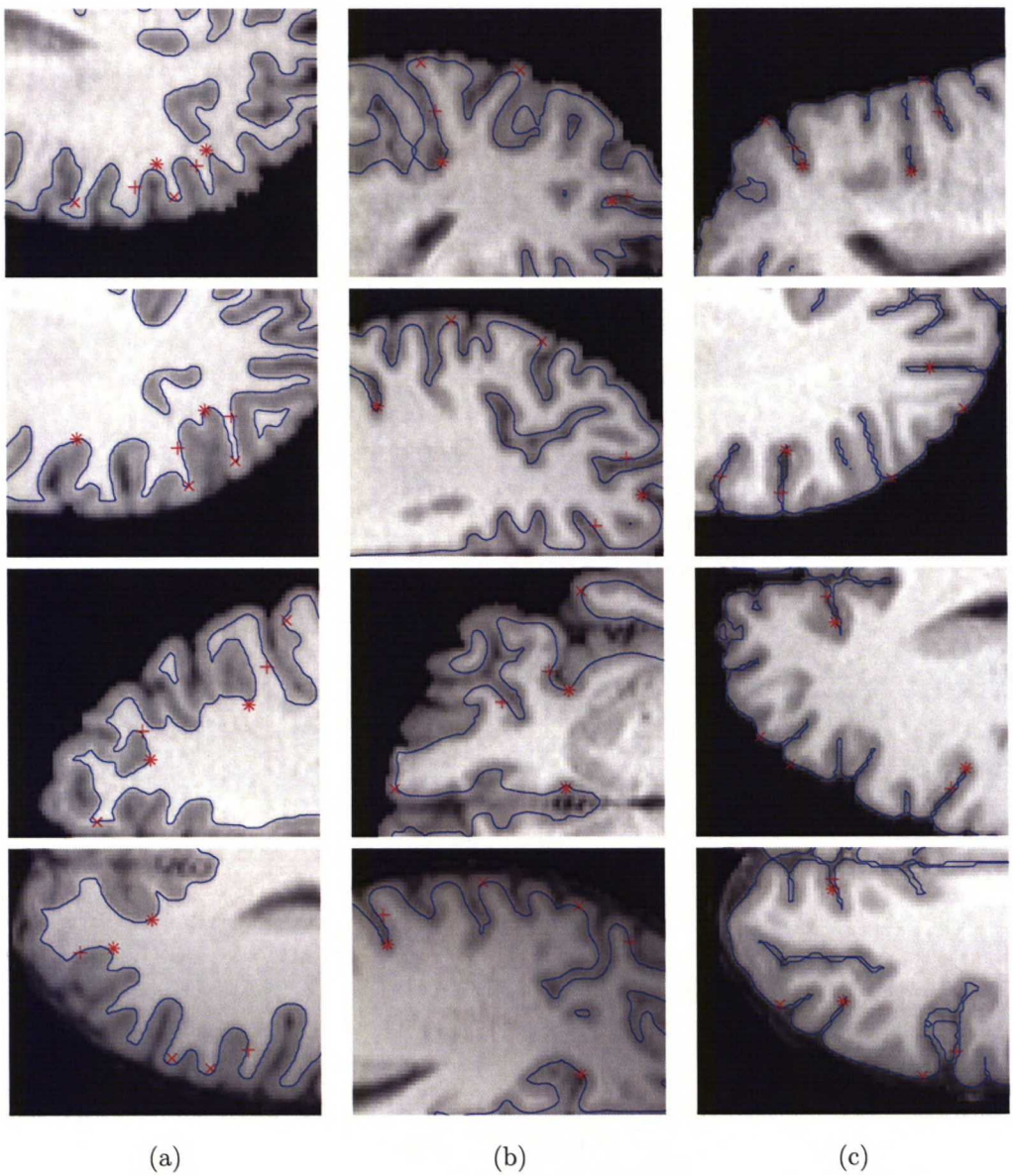


Figure 4.7 *Examples of selected landmark points on four subjects with one subject on each row, (a) inner surface, (b) central surface, (c) outer surface. Points selected as gyral crowns marked with "X", sulcal banks with "+" and sulcal fundi with "∗". The reconstructed surfaces are plotted with a blue line for reference (they were not visible when the landmark points were selected).*

of the central cortical surface, and Han et al. [24] used only five volumes with 10 landmarks for each brain.

One of the largest differences in results are in the estimated bias for the inner cortical surface reconstruction, where an outward bias was found in our study. Also, compared to the previous studies, a larger number of gross errors were found for the outer cortical surface, and the percentage of errors larger than 1 mm is significantly higher for the inner and central cortical surfaces in our study.

Table 4.1 *Summary of landmark study results, mean values for all three surfaces: signed distance (SD), absolute distance (AD) and proportion of absolute distance measures larger than values 1.0 and 2.0. Distances shown in millimeters together with corresponding standard deviations (std) as [distance ± std]*

Surface	Geometry	SD	AD	>1	>2
Inner	Crown	−0.53 ± 0.65	0.67 ± 0.50	23.23%	1.67%
	Bank	−0.52 ± 0.58	0.56 ± 0.54	12.29%	1.35%
	Fundus	−0.64 ± 0.57	0.67 ± 0.54	16.56%	1.35%
	All	−0.57 ± 0.60	0.63 ± 0.53	17.36%	1.46%
Central	Crown	−0.00 ± 0.93	0.73 ± 0.57	26.88%	3.65%
	Bank	−0.34 ± 0.70	0.58 ± 0.52	15.00%	1.56%
	Fundus	0.25 ± 0.89	0.70 ± 0.60	23.65%	3.33%
	All	−0.03 ± 0.88	0.67 ± 0.57	21.84%	2.85%
Outer	Crown	−0.10 ± 0.53	0.33 ± 0.43	3.96%	1.25%
	Bank	−0.09 ± 0.59	0.38 ± 0.46	5.63%	1.35%
	Fundus	−0.28 ± 0.95	0.53 ± 0.83	11.88%	3.02%
	All	−0.16 ± 0.72	0.42 ± 0.61	7.15%	1.88%
	Total	−0.25 ± 0.78	0.57 ± 0.58	15.45%	2.06%

Table 4.2 *Comparison of landmark study results: signed distance (SD), absolute distance (AD) and proportion of absolute distance measures larger than values 1.0 and 2.0. Distances shown in millimeters together with corresponding standard deviations (std) as [distance ± std]. Also shown landmarks per volume (L/Vol.) and number of volumes used (Vols.)*

Surface	Study	SD	AD	>1	>2	L/Vol.	Vols.
Inner	Han [24]	-	0.46 ± -	<12%	<1%	420*	2
	Tosun [56]	0.04 ± 0.73	0.50 ± 0.54	10%	3.1%	330*	2
	Häme	−0.57 ± 0.60	0.63 ± 0.53	17.36%	1.46%	96	30
Central	Han [24]	-	0.51 ± 0.41	6%	2%	10	5
	Tosun [56]	-	-	-	-	-	-
	Häme	−0.03 ± 0.88	0.67 ± 0.57	21.84%	2.85%	96	30
Outer	Han [24]	-	0.5 ± -	<12%	<0.3%	420*	2
	Tosun [56]	−0.09 ± 0.52	0.40 ± 0.34	6%	0.1%	330*	2
	Häme	−0.16 ± 0.72	0.42 ± 0.61	7.15%	1.88%	96	30

*Landmark points were selected in sets of 10 from each location.

Chapter 5

Discussion

5.1 Assessment of Results

The conducted tests show that the implementation of the reconstruction method was successful and produced expected results. Here the results are discussed together with some discovered problems and possible solutions for them. This chapter also includes an assessment of the applied evaluation methods followed by final conclusions and suggestions for possible directions of future work.

In preprocessing, problems were encountered at the skull-stripping stage. A fully automatic and reliable skull-stripping method would improve the preprocessing stage considerably.

The appropriate smoothing term in the fuzzy clustering depends strongly on the data quality, especially on the degree of noise present. To perform the fuzzy clustering robustly for images from different sources, a data-dependent and adaptive smoothing term would be necessary. Otherwise the clustering results have to be observed manually when confronting images from a new source.

The cortical fold opening stage showed locally unwanted results in some cases, which can be a significant problem if this occurs in an area of interest. This happens mostly at wide areas of GM where several fronts meet in the Fast Marching stage of the algorithm. In wide areas fronts originating from different locations meet at several locations causing numerous shock points and effectively low values for the gradient. This has the effect of reducing GM membership values in an area larger than desired. Also some displacement of the editing was noticed in some cases, but this seems to be a natural consequence of data with low contrast.

As noted in the previous chapter, the landmark study showed outward bias in the case of inner cortical surface. This might be solved simply by increasing the target membership value of the geometric deformable model, but adjusting the parameter values of the fuzzy clustering could also alleviate these problems. For example, by changing the fuzziness coefficient to allow more uncertainty in the clustering could add to the accuracy of the geometric deformable model. With the current parameter values for clustering the results have com-

monly a relatively steep transition between different classes, but it is important to keep in mind that changing these parameter values would also affect the two other reconstructions in a way that is impossible to predict. Contrary to our study, Tosun et al. [56] did not find bias for the inner cortical surface in their study. This distinction is most likely due to differences in landmark picking and amount of data used for the study, since the reconstruction methods were practically the same.

The central cortical surface had a similar error on average as the inner cortical surface. However, the central cortical had double the amount of gross errors. This means that the central surface reconstruction was accurate on average, but large errors also occurred, especially at gyral crowns and sulcal fundi. These may also be due to the difficulty of visually defining the location of the "true" central surface from the image. The central surface reconstruction was most accurate at sulcal banks.

The notion that the reconstruction was most accurate on the outer cortical surface is a surprising one, since the outer surface reconstruction can have difficulties with tight cortical folds. Even though the reconstruction of the outer surface was expectedly least accurate at sulcal fundi, the low average error results indicate that the reconstruction can find deep cortical folds with high accuracy in general. However, since the proportion of errors larger than 2 mm was notably large, this means that some deep cortical folds were still not located by the method, or that a local failure of the cortical fold opening displaced the reconstruction from the true position of the sulcal fundus.

Tosun also concluded that the parameter values of the geometric deformable model should be a function of cortical geometry, since different geometries achieve the highest accuracy with different values. This kind of study was not included in our work, however clear differences in the accuracy between different geometries were found, as reported above.

Comparison of results from different studies shows that the results obtained in this work are mostly consistent with the previous studies, even with differences in methods of landmark picking and the number of volumes used. For the central cortical surface landmark accuracy study, our work is by far the most extensive one, with 30 volumes and a total of 2880 landmark points.

5.2 Assessment of Evaluation Methods

The landmark study gives a somewhat good idea of the average accuracy, but as it never encloses the whole image volume, it is not in general able to notice if the reconstruction has failed locally. It is important to note that a full manual segmentation would have given more accurate error estimates for the entire image volumes, but unfortunately no accurate manual segmentations were available. Many studies use the IBSR data mentioned earlier, which includes manual segmentations of the image volumes. However, the manual segmentations do not distinguish cortical folds, and all voxels were merely marked as gray matter at locations where cortical folds were apparent. This fact made the data useless for our study, since one of the greatest strengths of the implemented method is its ability to find cortical folds.

Another problem associated with the landmark study is that it is possible to choose the landmark points inaccurately, without any effect on the error estimate. This is mainly associated with the points at sulcal fundi on the outer surface. If the landmark point is chosen "safely" as for some points that are not deep enough at the sulcal fundus in examples of figure 4.7(c), the error estimate can still be small since the surface passes close to the point. On the other hand, if the reconstructed surface is located too deep, inside the cortex, there is no way to see this from the results. Based on these observations, it is reasonable to say that the landmark measures for the outer surface at sulcal fundi are a measure whether the surface is "deep enough". In a similar way, the same problem is associated at gyral crowns for the inner surface.

Choosing the landmark points for the central surface was difficult, since there is no transition in image intensity at the center of the cortex. Still the error measures for the central surface were of similar magnitude as for the other surfaces.

Landmark selecting was done by only a single rater, which naturally causes bias always associated with a human rater. Bias is also caused by the fact that the rater had seen several reconstruction results before performing the landmark picking. This might affect the rater, however unintentionally, to choose points that are probable or improbable for the method to find.

The program used for landmark picking had some shortcomings that probably had effects on the results. The rater saw only a single image slice at a time, so image information in one direction was totally missing. Moreover, the selected points were not modifiable, and this caused a few poor initial choices to affect the results.

5.3 Conclusions and Future Work

In this work a cortical reconstruction method was implemented and evaluated with an unprecedentedly large data set. A fully automatic and independent software was successfully developed, taking fuzzy clustering memberships and the initial surface as input and generating three cortical surfaces as output. The software retains the initial topology for all surfaces and finds deep cortical folds that are typically difficult to find. The conducted extensive study using 8640 landmark points for 90 surface reconstructions from 30 images and visual inspection of reconstructions from 82 images show that the method is robust and accurate, achieving subvoxel accuracy on average. This conclusion was also reached in previous studies that our results were compared to. The inner surface reconstruction showed outward bias in our study. The method has a low rate of gross errors, although the possibility of large errors in critical locations is not totally eliminated. In addition, also an interactive method for ventricle filling was developed and used for all reconstructions. Other stages of the reconstruction method were performed with publicly available tools.

In the future, we will apply this effective method for segmentation and reconstruction to other purposes and image modalities with appropriate modifications. For cortex reconstruction, there are also several possible areas of improvement to remove manual interaction and improve robustness and accuracy.

The automatic and independent software could be extended to cover the stages performed currently with publicly available tools. This would render the software fully independent and devoid of any manual operation. The skull-stripping in the preprocessing stage could be replaced with an automatic method that takes intensity inhomogeneities into consideration. The topology correction stage could be changed to use gray level values of the membership function instead of the binarized volume. This should cause the algorithm to cut handles at more favorable locations, since also the membership values are at its disposal. This should subsequently increase the accuracy of the geometric deformable model initialization. For the geometric deformable model to reach higher accuracy depending on cortical geometry, adaptive parameters could be chosen through analysis of the mean curvature on the surface, which is already used in the implicit surface evolution.

Different options can be investigated for preventing editing unwanted areas of the gray matter membership in opening cortical folds. The ventricle filling method should be improved to eliminate manual editing and make this stage fully automatic as well. Robustness could be improved by restraining the filling from advancing outside the ventricular system. Both of these goals are difficult to achieve without using some prior model for the ventricles, so this should be one option to be considered. For future studies the landmark selection program can be developed to achieve results more reliably. The landmark accuracy study could also be conducted with several raters to minimize rater-dependent bias.

Bibliography

[1] Adalsteinsson, D., Sethian, J.A., 1995. A fast level set method for propagating interfaces. Journal of Computational Physics, vol. 118, pp. 269–277.

[2] Awate, S.P., Tasdizen, T., Foster, N., Whitaker, R.T., 2006. Adaptive, nonparametric Markov modeling for unsupervised, MRI brain-tissue classification. Preprint: under review at Medical Image Analysis (MedIA) Journal. Scientific Computing and Imaging Institute, University of Utah.

[3] Baillard, C., Hellier, P., Barillot, C., 2001. Segmentation of brain 3D MR images using level sets and dense registration. Medical Image Analysis, vol. 5, no.3 pp. 185–194. Elsevier Science B.V.

[4] Baillet S., Garnero, L., 1997. A Bayesian approach to introducing anatomo-functional priors in the EEG/MEG inverse problem. IEEE Transactions on Biomedical Engineering, vol. 44, no. 5.

[5] Bazin, P.L., Pham, D.L., 2007. Topology correction of segmented medical images using a fast marching algorithm. Computer methods and programs in biomedicine, vol. 88. pp. 182–190.

[6] Bazin, P.L., Pham, D.L., 2007. Topology-preserving tissue classification of magnetic resonance brain images. IEEE Transactions on Medical Imaging, Vol. 26, No. 4.

[7] Belaroussi, B., Milles, J., Carme, S., Zhu, Y.M., Benoit-Cattin, H., 2006. Intensity non-uniformity correction in MRI: Existing methods and their validation. Medical Image Analysis, vol. 10. Elsevier B.V.

[8] Bertrand, G., 1994. Simple points, topological numbers and geodesic neighborhoods in cubic grids. Pattern Recognition Letters, vol. 15, pp. 1003–1011.

[9] Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.

[10] Caselles, V., Kimmel, R., Sapiro, G., 1997. Geodesic active contours. International Journal of Computer Vision, vol. 22, no. 1. Kluwer Academic Publishers.

[11] Clarke, L.P., Velthuizen, R.P., Camacho, M.A., Heine, J.J., Vaidyanathan, M., Hall, L.O., Thatcher, R.W., Silbiger, M.L., 1995. MRI segmentation: methods and applications. Magnetic Resonance Imaging, vol. 13, no. 3. Elsevier Science Ltd.

[12] Cocosco, C.A., Kollokian, V., Kwan, R.K.S., Evans, A.C., 1997. BrainWeb: online interface to a 3D MRI simulated brain database. NeuroImage, vol. 5, no. 4. Online: http://www.bic.mni.mcgill.ca/brainweb/ [referred 19.2.2008]

[13] Cocosco, C.A., Zijdenbos, A.P., Evans, A.C., 2003. A fully automatic and robust brain MRI tissue classification method. Medical Image Analysis, vol. 7.

[14] Cuadra, M.B., Cammoun, L., Butz, T., Cuisenaire, O., Thiran, J.P., 2005. Comparison and validation of tissue modelization and statistical classification methods in T1-weighted MR brain images. IEEE Transactions on Medical Imaging, vol. 24, no. 12.

[15] Dale, A.M., Fischl, B., Sereno, M.I., 1999. Cortical surface-based analysis I. Segmentation and surface reconstruction. NeuroImage, vol. 9, no. 2, pp. 179–194. Academic Press.

[16] Dokládal, P., Bloch, I., Couprie, M., Ruijters, D., Urtasun, R., Garnero, L., 2003. Topologically controlled segmentation of 3D magnetic resonance images of the head by using morphological operators. Pattern recognition 36. Elsevier Ltd.

[17] Dowsett, D.J., Kenny, P.A., Johnston, R.E., 1998. The Physics of Diagnostic Imaging. Chapman and Hall.

[18] Eskildsen, S.F., Ostergaard, L.R., 2006. Active surface approach for extraction of the human cerebral cortex from MRI. MICCAI 2006. Springer-Verlag Berlin Heidelberg.

[19] Fischl, B., Salat, D.H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., van der Kouwe, A., Killiany, R., Kennedy, D., Klaveness, S., Montillo, A., Makris, N., Rosen, B., Dale, A.M., 2002. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. Neuron, vol. 33. Cell Press.

[20] Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M., 2002. Cortex segmentation: a fast variational geometric approach. IEEE Transactions on Medical Imaging, vol. 21, no. 2.

[21] Goldszal, A.F., Davatzikos, C., Pham, D.L., Yan, M.X.H., Bryan, R.N., Resnick, S.M., 1998. An image processing system for qualitative and quantitative volumetric analysis of brain images. Journal of Computer Assisted Tomography, vol. 22, no. 5, pp. 827–837.

[22] Greenspan, H., Ruf, A., Goldberger, J., 2006. Constrained gaussian mixture model framework for automatic segmentation of MR brain images. IEEE Transactions on Medical Imaging, vol. 25, no. 9.

[23] Han, X., Xu, C., Prince, J.L., 2003. A topology preserving level set method for geometric deformable models. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 6.

[24] Han, X., Pham, D.L., Tosun, D., Rettmann, M.E., Xu, C., Prince, J.L., 2004. CRUISE: Cortical reconstruction using implicit surface evolution. NeuroImage, vol. 23, pp. 997–1012.

[25] Han, X., Xu, C., Rettmann, M.E., Prince, J.L., 2001. Automatic segmentation editing for cortical surface reconstruction. Proceedings of SPIE Medical Imaging, vol. 4322, pp. 194–203. SPIE Press, Bellingham, WA.

[26] Held, K., Kops, E.R., Krause, B.J., Wells, W.M., Kikinis, R., Müller-Gärtner, H.W., 1997. Markov random field segmentation of brain MR images. IEEE Transactions on Medical Imaging, vol. 16, no. 6.

[27] Ibrahim, M., John, N., Kabuka, M., Younis, A., 2006. Hidden Markov models-based 3D MRI brain segmentation. Image and Vision Computing 24. Elsevier B.V.

[28] Jiménez-Alaniz, J.R., Medina-Banuelos, V., Yánez-Suárez, O., 2006. Data-Driven brain MRI segmentation supported on edge confidence and a priori tissue information. IEEE Transactions on Medical Imaging, vol. 25, no. 1.

[29] Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. International Journal of Computer Vision, vol. 1, pp. 312–333.

[30] Kong, J., Wang, J., Lu, Y., Zhang, J., Li, Y., Zhang, B., 2006. A novel approach for segmentation of MRI brain images. IEEE Melecon.

[31] Li, H., Yezzi, A., Cohen, L.D., 2006. 3D brain segmentation using dual-front active contours with optional user interaction. International Journal of Biomedical Imaging, vol. 2006. Hindawi Publishing Corporation.

[32] Liew, A.W.C., Yan, H., 2006. Current methods in the automatic tissue segmentation of 3D magnetic resonance brain images. Current Medical Imaging Reviews. Bentham Science Publishers Ltd.

[33] MacDonald, D., Kabani, N., Avis, D., Evans, A.C., 2000. Automated 3D extraction of inner and outer surfaces of cerebral cortex from MRI. Neuroimage, vol. 12. Academic Press.

[34] Malladi, R., Sethian, J.A., Vemuri, B.C., 1995. Shape modeling with front propagation: a level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 2.

[35] Marroquin, J.L., Vemuri, B.C., Botello, S., Calderon, F., Fernandez-Bouzas, A., 2002. An accurate and efficient bayesian method for automatic segmentation of brain MRI. IEEE Transactions on Medical Imaging, vol. 21, no. 8.

[36] McAuliffe M.J., Lalonde F.M., McGarry D., Gandler W., Csaky K., Trus B.L., 2001. Medical image processing, analysis and visualization in clinical research. IEEE Computer-based Medical Systems (CBMS), pp. 381–386.

[37] Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics, vol. 79.

[38] Osher, S.J., Fedkiw, R., 2003. Level Set Methods and Dynamic Implicit Surfaces. Springer-Verlag New York, Incorporated.

[39] Pham, D.L., Xu, C., Prince, J.L., 1998. A survey of current methods in medical image segmentation. Annual Review of Biomedical Engineering, 1998.

[40] Pham, D.L., Prince, J.L., 1999. Adaptive fuzzy segmentation of magnetic resonance images. IEEE Transaction on Medical Imaging, vol. 18, no. 9.

[41] Pham, D.L., 2001. Robust fuzzy segmentation of magnetic resonance images. Proceedings of the Fourteenth IEEE Symposium on Computer-Based Medical Systems (CBMS2001). IEEE Press, Somerset, NJ, pp. 127–131.

[42] Pham, D.L., 2001. Spatial models for fuzzy clustering. Computer Vision and Image Understanding, vol. 84, pp. 285–297.

[43] Rajapakse, J.C., Giedd, J.N., Rapoport, J.L., 1997. Statistical approach to segmentation of single-channel cerebral MR images. IEEE Transactions on Medical Imaging, vol. 16, no. 2.

[44] Ruan, S., Moretti, B., Fadili, J., Bloyet, D., 2002. Fuzzy Markovian segmentation in application of magnetic resonance images. Computer Vision and Image Understanding, vol. 85. Elsevier Science.

[45] Sedgewick, R., 1988. Algorithms. Second edition. Addison-Wesley Publishing Company, Inc.

[46] Segonne, F., Pons, J.P. Fischl, B., Grimson, E., 2005. A novel active contour framework: multicomponent level set evolution under topology control. Massachusetts Institute of Technology.

[47] Sethian, J.A., 1996. A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences, vol. 93, pp. 1591–1595.

[48] Sethian, J.A., 1999. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science. Cambridge University Press, Cambridge, UK.

[49] Shattuck, D.W., Sandor-Leahy, S.R., Schaper, K.A., Rottenberg, D.A., Leahy, R.M., 2000. Magnetic resonance image tissue classification using a partial volume model. University of Southern California.

[50] Shattuck, D.W., Leahy, R.M., 2002. BrainSuite: An automated cortical surface identification tool. Medical Image Analysis, vol. 6, no. 2, pp. 129–42.

[51] Suri, J.S., Singh, S., Reden, L., 2002. Computer vision and pattern recognition techniques for 2-D and 3-D MR cerebral cortical segmentation (Part I): A state-of-the-art review. Pattern Analysis and Applications, vol. 5, no. 1. Springer-Verlag Ltd.

[52] Suri, J.S., Singh, S., Reden, L., 2002. Fusion of region and boundary/surface-based computer vision and pattern recognition techniques for 2-D and 3-D MR cerebral cortical segmentation (Part II): A state-of-the-art review. Pattern Analysis and Applications, vol. 5, no. 1. Springer-Verlag Ltd.

[53] Theodoridis, S., Koutroumbas, K., 2006. Pattern Recognition. Third Edition. Elsevier (USA).

[54] Tortora, G.J., Grabowski, S.R., 2000. Principles of Anatomy and Physiology. John Wiley and Sons, inc.

[55] Tosun, D., Rettmann, M.E., Han, X. Tao, X., Xu, C., Resnick, S.M., Pham, D.L. Prince, J.L., 2004. Cortical surface segmentation and mapping. NeuroImage, vol. 23. Elsevier Inc.

[56] Tosun, D., Rettmann M.E., Naiman, D.Q., Resnick, S.M., Kraut, M.A., Prince, J.L., 2005. Cortical reconstruction using implicit surface evolution: Accuracy and precision analysis. NeuroImage, vol. 29, pp. 838–852.

[57] Van Essen, D.C., Dickson, J., Harwell, J., Hanlon, D., Anderson, C.H. and Drury, H.A., 2001. An integrated software system for surface-based analyses of cerebral cortex. Journal of American Medical Informatics Association, online: http://brainmap.wustl.edu/caret [referred 19.2.2008]

[58] Van Leemput, K., Maes, F., Vandermeulen, D., Suetens, P., 1999. Automated model-based tissue classification of MR images of the brain. IEEE Transactions on Medical Imaging, vol. 18, no. 10.

[59] Van Leemput, K. Expectation-maximization segmentation. Fully automated model-based segmentation of MR images of the brain. Katholieke Universiteit Leuven. Online: http://mic.uzleuven.be/MedicalImageComputing/downloads/ems.php [referred 19.2.2008]

[60] Wang, H., Sturm, P., Schmitt, F., Priese, L., 2006. Improved segmentation of MR brain images including bias field correction based on 3D-CSC. Institute for Computational Visualistics, University of Koblenz-Landau.

[61] Xu, C., Prince, J.L., 1998. Snakes, shapes and gradient vector flow. IEEE Transactions on Image Processing, vol. 7, no. 3.

[62] Xu, C., Prince, J.L., 1998. Generalized gradient vector flow external forces for active contours. Signal Processing, vol. 71, no. 2, pp. 132–139.

[63] Xu, C. Pham, D.L., Rettmann, M.E., Yu, D.N., Prince, J.L., 1999. Reconstruction of the human cerebral cortex from magnetic resonance images. IEEE Transactions on Medical Imaging, vol. 18, no. 6.

[64] Zeng, X., Staib, L.H., Schultz, R.T., Duncan J.S., 1999. Segmentation and measurement of the cortex from 3-D MR images using coupled-surfaces propagation. IEEE Transactions on Medical Imaging, vol. 18, no. 10.

[65] Zhang, Y., Brady, M., Smith, S., 2001. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Transactions on Medical Imaging, vol. 20, no. 1.

Appendix A

Results

Table A.1 *Landmark study results for inner surface, mean values by subject: signed distance (SD), absolute distance (AD) and proportion of absolute distance measures larger than values 1.0 and 2.0. Distances shown in millimeters together with corresponding standard deviations (std) as [distance ± std]*

Subject	SD	AD	AD > 1	AD > 2
1	−0.76 ± 0.54	0.78 ± 0.51	29.17%	2.08%
2	−1.01 ± 0.51	1.02 ± 0.48	50.00%	3.13%
3	−0.77 ± 0.56	0.84 ± 0.45	28.13%	2.08%
4	−0.68 ± 0.65	0.70 ± 0.63	17.71%	3.13%
5	−0.80 ± 0.56	0.82 ± 0.54	27.08%	5.21%
6	−0.63 ± 0.48	0.68 ± 0.40	17.71%	1.04%
7	−0.19 ± 0.41	0.36 ± 0.27	3.13%	0.00%
8	−0.32 ± 1.03	0.52 ± 0.94	7.29%	2.08%
9	−0.39 ± 0.37	0.44 ± 0.31	6.25%	0.00%
10	−0.30 ± 0.44	0.39 ± 0.36	6.25%	0.00%
11	−0.32 ± 0.75	0.50 ± 0.65	8.33%	2.08%
12	−0.28 ± 0.50	0.47 ± 0.32	6.25%	0.00%
13	−0.50 ± 0.50	0.56 ± 0.43	13.54%	0.00%
14	−0.50 ± 0.40	0.54 ± 0.34	9.38%	0.00%
15	−0.79 ± 0.56	0.84 ± 0.49	33.33%	1.04%
16	−0.77 ± 0.62	0.84 ± 0.52	36.46%	2.08%
17	−0.41 ± 0.50	0.54 ± 0.36	12.50%	0.00%
18	−0.50 ± 0.41	0.54 ± 0.36	11.46%	0.00%
19	−0.45 ± 0.44	0.52 ± 0.35	6.25%	0.00%
20	−0.79 ± 0.50	0.82 ± 0.46	30.21%	2.08%
21	−0.34 ± 0.34	0.38 ± 0.29	3.13%	0.00%
22	−0.92 ± 0.69	0.94 ± 0.65	41.67%	7.29%
23	−0.44 ± 0.28	0.45 ± 0.26	1.04%	0.00%
24	−0.85 ± 0.87	0.85 ± 0.86	26.04%	3.13%
25	−0.60 ± 0.69	0.73 ± 0.55	26.04%	3.13%
26	−0.54 ± 0.41	0.60 ± 0.32	12.50%	0.00%
27	−0.76 ± 0.82	0.77 ± 0.81	17.71%	3.13%
28	−0.39 ± 0.37	0.45 ± 0.29	3.13%	0.00%
29	−0.73 ± 0.47	0.74 ± 0.46	23.96%	1.04%
30	−0.19 ± 0.42	0.33 ± 0.32	5.21%	0.00%
All	−0.57 ± 0.60	0.63 ± 0.53	17.36%	1.46%

Table A.2 *Landmark study results for central surface, mean values by subject: signed distance (SD), absolute distance (AD) and proportion of absolute distance measures larger than values 1.0 and 2.0. Distances shown in millimeters together with corresponding standard deviations (std) as [distance ± std]*

Subject	SD	AD	AD > 1	AD > 2
1	−1.17 ± 0.97	1.23 ± 0.90	54.17%	16.67%
2	−0.98 ± 0.63	1.01 ± 0.57	43.75%	6.25%
3	−0.80 ± 1.12	1.03 ± 0.91	43.75%	7.29%
4	−0.15 ± 0.60	0.50 ± 0.35	10.42%	0.00%
5	−0.42 ± 0.68	0.65 ± 0.46	22.92%	0.00%
6	−0.21 ± 0.76	0.60 ± 0.51	16.67%	4.17%
7	0.12 ± 0.77	0.59 ± 0.50	16.67%	3.13%
8	1.11 ± 0.83	1.18 ± 0.74	51.04%	13.54%
9	0.07 ± 0.66	0.51 ± 0.43	9.38%	1.04%
10	0.05 ± 0.57	0.45 ± 0.35	10.42%	0.00%
11	0.14 ± 0.95	0.68 ± 0.68	23.96%	2.08%
12	0.33 ± 0.62	0.52 ± 0.47	16.67%	0.00%
13	−0.06 ± 0.69	0.56 ± 0.41	11.46%	0.00%
14	0.18 ± 0.57	0.46 ± 0.38	7.29%	0.00%
15	−0.47 ± 0.88	0.81 ± 0.58	30.21%	5.21%
16	−0.05 ± 0.78	0.62 ± 0.47	19.79%	1.04%
17	0.18 ± 0.67	0.54 ± 0.42	12.50%	0.00%
18	0.36 ± 0.77	0.60 ± 0.60	14.58%	4.17%
19	0.14 ± 0.76	0.65 ± 0.43	20.83%	1.04%
20	−0.29 ± 0.89	0.73 ± 0.58	31.25%	1.04%
21	0.44 ± 0.66	0.63 ± 0.48	20.83%	1.04%
22	−0.24 ± 0.82	0.68 ± 0.52	23.96%	1.04%
23	0.37 ± 0.66	0.60 ± 0.46	20.83%	0.00%
24	−0.20 ± 0.99	0.80 ± 0.60	29.17%	7.29%
25	−0.04 ± 0.73	0.58 ± 0.44	15.63%	1.04%
26	0.25 ± 0.71	0.60 ± 0.45	15.63%	1.04%
27	−0.02 ± 0.58	0.48 ± 0.33	6.25%	0.00%
28	0.39 ± 0.75	0.66 ± 0.53	23.96%	2.08%
29	−0.28 ± 0.70	0.59 ± 0.47	13.54%	2.08%
30	0.37 ± 0.70	0.60 ± 0.51	17.71%	3.13%
All	−0.03 ± 0.88	0.67 ± 0.57	21.84%	2.85%

Table A.3 *Landmark study results for outer surface, mean values by subject: signed distance (SD), absolute distance (AD) and proportion of absolute distance measures larger than values 1.0 and 2.0. Distances shown in millimeters together with corresponding standard deviations (std) as [distance ± std]*

Subject	SD	AD	AD > 1	AD > 2
1	−0.60 ± 0.76	0.68 ± 0.69	17.71%	3.13%
2	−0.83 ± 1.20	0.87 ± 1.16	27.08%	4.17%
3	−0.68 ± 1.09	0.77 ± 1.03	15.63%	5.21%
4	−0.19 ± 0.52	0.35 ± 0.43	6.25%	2.08%
5	−0.15 ± 0.41	0.30 ± 0.31	2.08%	0.00%
6	−0.16 ± 0.34	0.29 ± 0.24	2.08%	0.00%
7	−0.10 ± 0.62	0.32 ± 0.54	6.25%	2.08%
8	0.85 ± 1.26	1.15 ± 1.00	46.88%	13.54%
9	0.04 ± 0.35	0.28 ± 0.22	2.08%	0.00%
10	0.03 ± 0.40	0.28 ± 0.29	2.08%	1.04%
11	−0.13 ± 0.64	0.36 ± 0.55	5.21%	1.04%
12	0.10 ± 0.34	0.28 ± 0.21	1.04%	0.00%
13	−0.42 ± 1.24	0.57 ± 1.18	11.46%	7.29%
14	0.02 ± 0.30	0.23 ± 0.20	0.00%	0.00%
15	−0.07 ± 0.37	0.31 ± 0.22	0.00%	0.00%
16	−0.15 ± 0.49	0.35 ± 0.38	4.17%	2.08%
17	−0.02 ± 0.37	0.27 ± 0.26	3.13%	0.00%
18	−0.12 ± 0.63	0.33 ± 0.55	4.17%	2.08%
19	−0.28 ± 0.63	0.50 ± 0.48	7.29%	1.04%
20	−0.27 ± 0.95	0.45 ± 0.88	8.33%	2.08%
21	−0.02 ± 0.39	0.30 ± 0.24	1.04%	0.00%
22	−0.22 ± 0.47	0.38 ± 0.36	8.33%	0.00%
23	−0.11 ± 0.38	0.28 ± 0.27	3.13%	0.00%
24	−0.27 ± 0.48	0.43 ± 0.35	6.25%	1.04%
25	−0.22 ± 0.57	0.41 ± 0.45	7.29%	2.08%
26	−0.11 ± 0.37	0.29 ± 0.25	2.08%	0.00%
27	−0.20 ± 0.59	0.35 ± 0.52	3.13%	2.08%
28	−0.10 ± 0.39	0.31 ± 0.25	0.00%	0.00%
29	−0.35 ± 1.04	0.49 ± 0.98	7.29%	4.17%
30	−0.09 ± 0.38	0.29 ± 0.26	3.13%	0.00%
All	−0.16 ± 0.72	0.42 ± 0.61	7.15%	1.88%

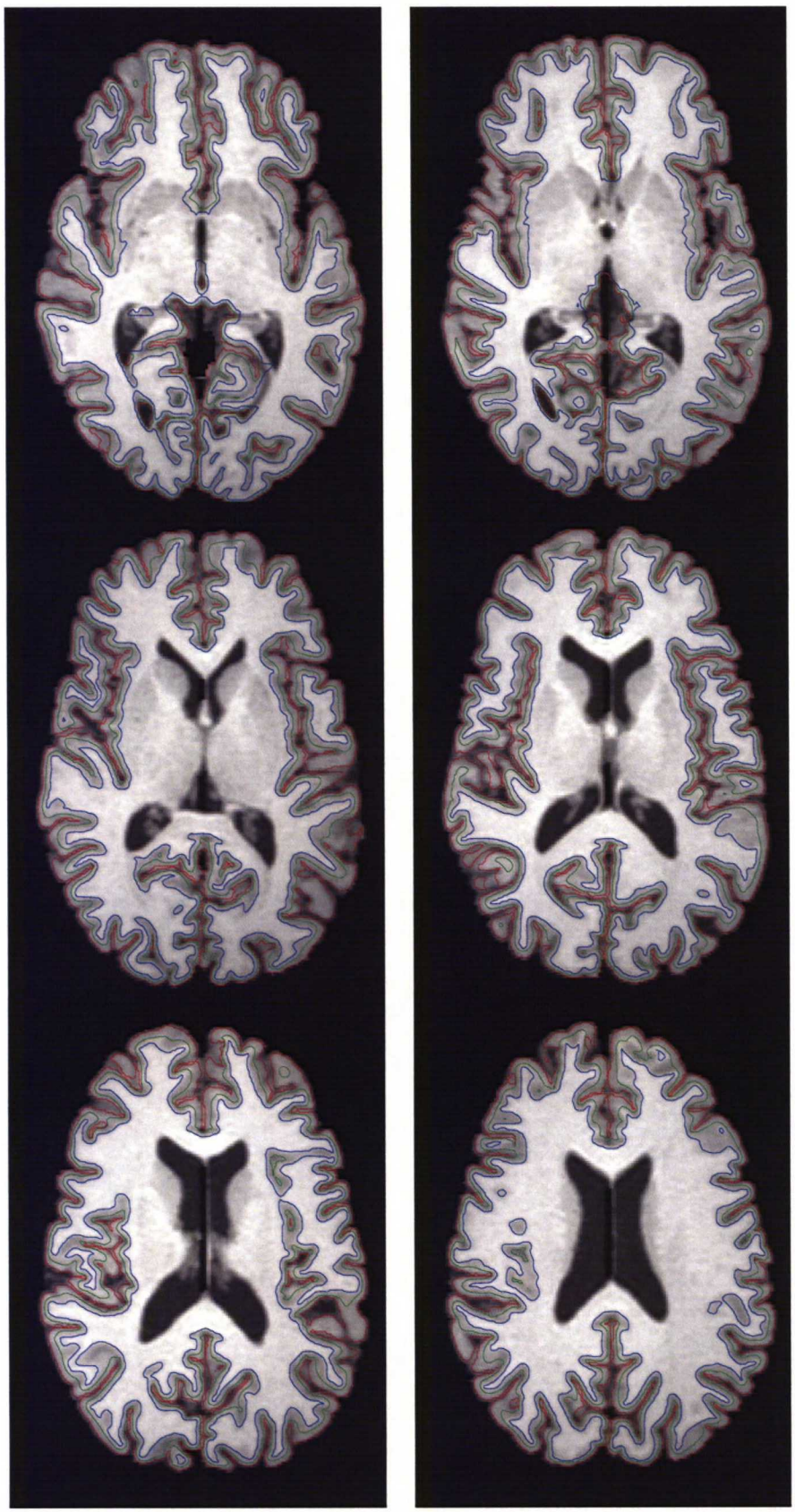


Figure A.1 *Example of reconstruction results overlaid on axial slices: IXI database subject 37, in intervals of five slices (blue: inner, green: central, red: outer)*

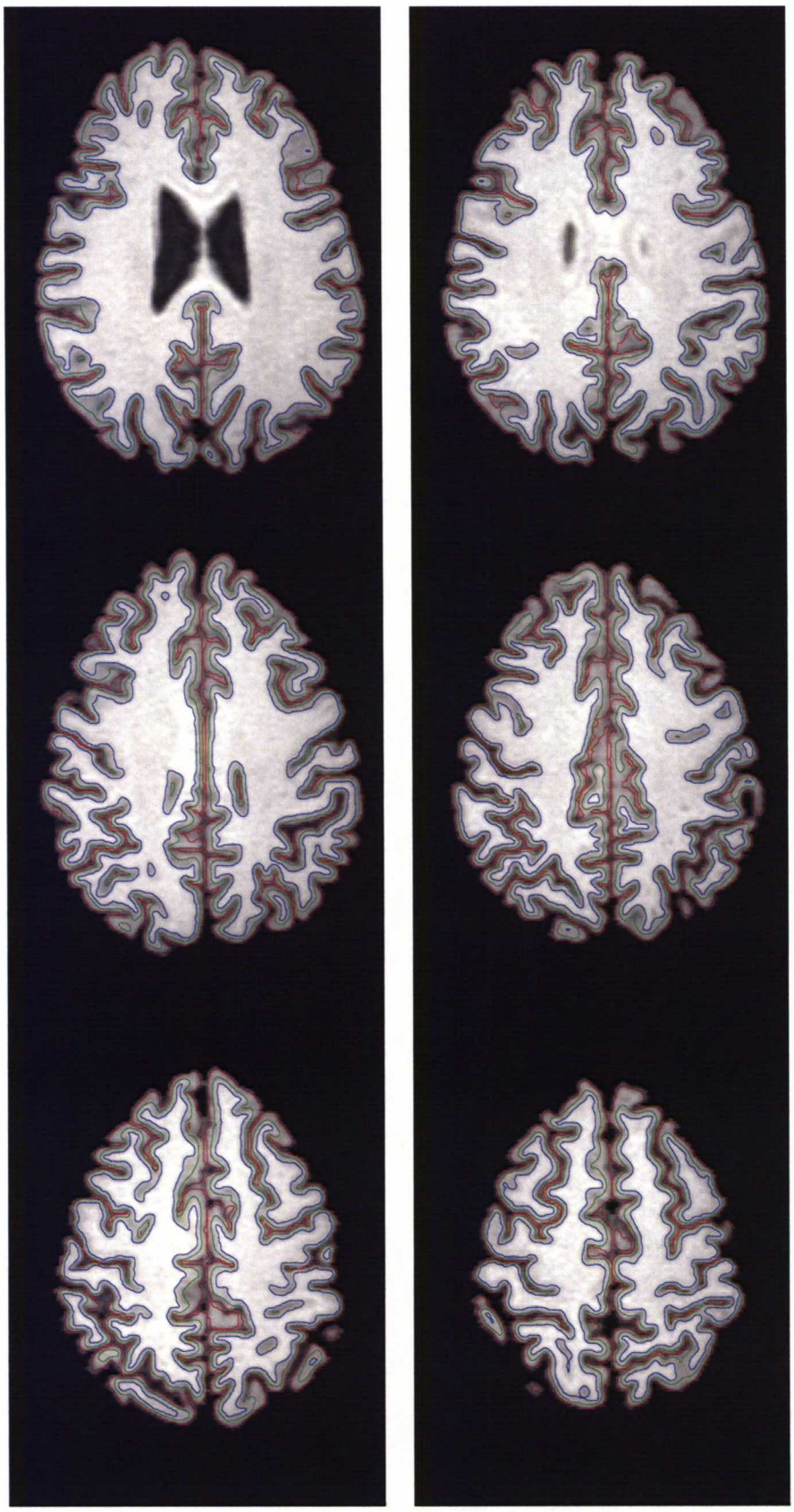


Figure A.2 *Example of reconstruction results overlaid on axial slices: IXI database subject 37, in intervals of five slices (blue: inner, green: central, red: outer)*

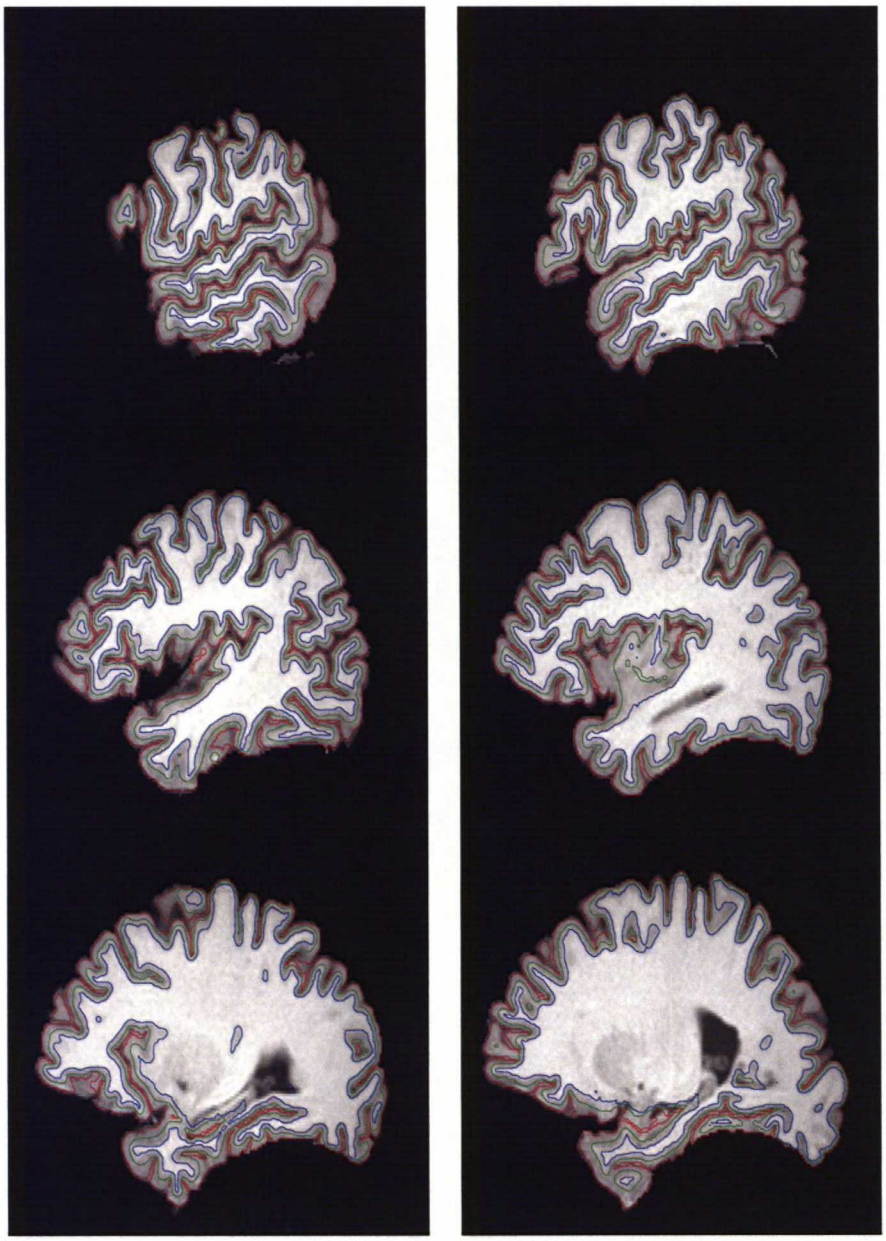


Figure A.3 *Example of reconstruction results overlaid on sagittal slices: IXI database subject 37, in intervals of three slices (blue: inner, green: central, red: outer).*

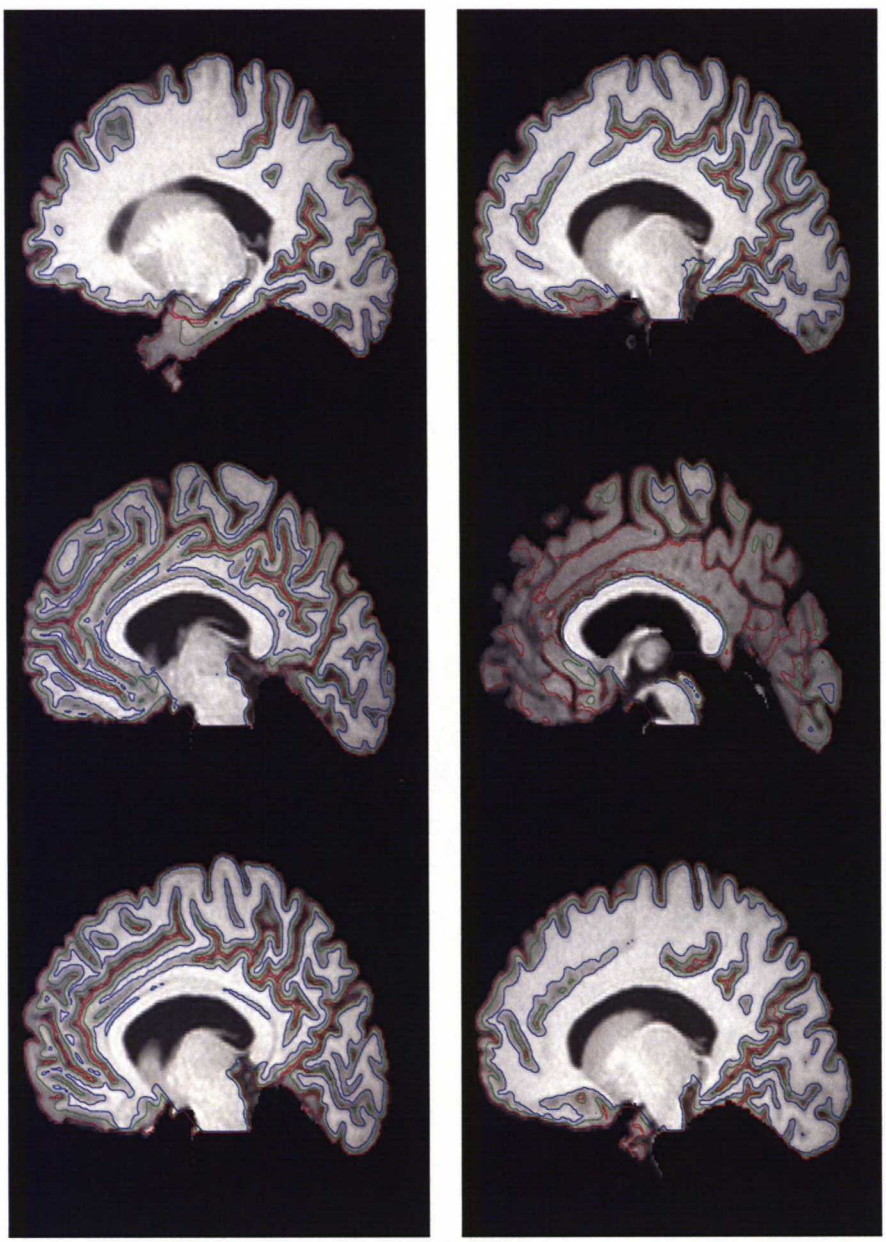


Figure A.4 *Example of reconstruction results overlaid on sagittal slices: IXI database subject 37, in intervals of three slices (blue: inner, green: central, red: outer)*

Appendix B

Tools

Several tools are available to help with the process of cortex segmentation. Here we briefly present some publicly available software for cortex segmentation and several other tasks related to segmentation, such as preprocessing or visualizing the image.

Already in the previous chapter helpful resources for evaluation of brain segmentation algorithms were mentioned, the synthetic brain MR image generator BrainWeb and IBSR, which offers real images segmented by an expert and by some basic algorithms.

FreeSurfer (online: http://surfer.nmr.mgh.harvard.edu/, [referred 19.2.2008]) is a freely available software for the study of cortical and subcortical anatomy. FreeSurfer can extract the boundaries of the cortex, preprocess MRI volumes and label subcortical classes. It offers a graphical user interface for data visualization, analysis and management. For the cortex surface extraction, FreeSurfer uses the method of Dale et al. [15], presented earlier in this work.

Another software for cortical surface extraction is the BrainSuite 2.0 [50], former versions known as BrainSuite and BSE (Brain Surface Extractor). BrainSuite 2.0 is a free, open-source software that offers an implementation of the method proposed by Shattuck et al. [49], with additional topology correction. The software also has many tools for preprocessing stages and a graphical user interface for user-friendly representation of different stages.

A recent addition to the list of programs is the Java-based Medical Image Processing, Analysis and Visualization (MIPAV) (see McAuliffe et al. [36]). In addition to the wide range of functionality provided by MIPAV, there are also useful plug-ins available for various operations.

A freely available suite of MATLAB functions and subroutines for automatic brain tissue segmentation for MR images by Van Leemput [59] is available online. This package is called Expectation-maximization segmentation (EMS) and it is an implementation of the method proposed by Van Leemput et al. [58].

Caret (Computerized Anatomical Reconstruction and Editing Toolkit) is a freely available program for segmentation, interactive viewing, manipulating and analyzing surface recon-

structions of the cerebral and cerebellar cortex by Van Essen et al. [57]. In addition, there is the commercially distributed program BrainVoyager (Cambridge Research Systems, online: http://www.crsltd.com/catalog/brainvoyager/, [referred 21.3.2008]) which includes a wide range of related functions, such as cortex segmentation, reconstruction, inflation and flattening.